Class 8: Weighted constraints II; lab

To do for Monday
- Reading: Guy 1991
  - Reading question: could any aspects of your own data be seen as belonging to different levels? E.g., are there different morphological environments in which some rule applies? Discuss why or why not briefly (about 1 page)

Overview: Another type of weighted grammar: Maximum Entropy OT. Smoothing/overfitting revisitied. Lab on MaxEnt.

1 Noisy HG and probabilities

- We saw that noisy Harmonic Grammar grammars are not probability distributions over classic OT grammars (ganging up, cumulativity)
  - However, they are still interpretable as a probability distribution over (infinitely many) non-noisy HG grammars:
  - A candidate's probability of winning, given certain weights, is the probability that the *weights+noise* select it as the winner

- Today, in our last type of quantitative constraint model (MaxEnt OT), we'll see a theory where the grammar assigns probabilities to candidates directly.

2 Multinomial logistic regression

- To get to MaxEnt OT, we need to go back to logistic regression.
- Let's try to predict whether a verb is irregular as a function of whether it ends with a nasal consonant and whether the last syllable begins with a complex onset (Lieberman & al. data modified)

==> Projector: I'll show you the modified input file

```
> summary(irregs.glm)

Call:
glm(formula = Modern_Irregular ~ nasal_in_stressed_coda +
stressed_onset_complex,
    family = binomial, data = irregulars)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)               0.7026     0.1775   3.957 7.58e-05 ***
nasal_in_stressed_coda    1.4504     0.4687   3.094  0.00197 **
stressed_onset_complex   -0.4060     0.2937  -1.383  0.16679
```

- This means: $probability(irregular) = \frac{1}{1+e^{-(0.7026+1.4504*nas-0.4060*complex)}}$
  - What's *probability(irregular)*?

- Now suppose we have a **multi-valued** dependent variable: "modern_type2", with values *devoice* (bend-bent), *other* (go-went), *regular* (talk-talked), *suffix_shorten* (feel-felt), *vowel* (sing-sang), *zero* (bet-bet)

We have to use a different function from `glm()`. I used `multinom()` here, in the `nnet` package (Venables & Ripley 2002)

```
> summary(irregs.multinom)
Call:
multinom(formula = modern_type2 ~ nasal_in_stressed_coda + stressed_onset_complex +
    ends_t_d, data = irregulars)

Coefficients:
              (Intercept) nasal_in_stressed_coda stressed_onset_complex    ends_t_d
devoice        -27.4591746              5.0364303             -2.3401159  25.2147595
other           -1.2613503              1.6038698             -1.3259646  -0.1535805
suffix_shorten  -1.4963996              0.6211568             -0.4098574 -24.2012397
vowel           -0.3154619              1.7520775             -0.3115235   0.8304717
zero           -16.9563332            -15.9293614             -0.6660790  17.7623345
```

- How to unpack this:
  - First line compares "devoice" to "regular" (I used the `relevel()` command to make "regular" the baseline—see today's R script, which I'll post)

$$\ln\left(\frac{prob(type=devoice)}{prob(type=regular)}\right) = -27.46 + 5.04 * nas - 2.34 * complex + 25.21 * t\_d$$

  - Second line:

$$\ln\left(\frac{prob(type=other)}{prob(type=regular)}\right) = -1.26 + 1.60 * nas - 1.33 * complex - 0.15 * t\_d$$

  - etc.
- Now we have to do some algebra on the board to find *prob(type=regular)* [hint: remember that the probabilities of the 6 choices must sum to 1]
- What we should get is:

$$prob(regular) = \frac{1}{1 + e^{linear\_regression\_for\_devoice} + e^{line\_for\_other} + e^{line\_suff\_shorten} + e^{line\_vowel} + e^{line\_zero}}$$

- Call the denominator in the expression above *Z*.
- Then $prob(devoice) = \frac{1}{Z} e^{line\_for\_devoice}$, $prob(other) = \frac{1}{Z} e^{line\_for\_other}$, etc.

This should start to remind you of theory you read about in Martin 2007!!

3 Maximum Entropy OT[1] (Goldwater & Johnson 2003)

- Goldwater & Johnson proposed applying to constraint grammars in linguistics a technique well-known in Machine Learning.
- Machine-Learning people tend to call it "Maximum entropy classification" instead of "multinomial logistic regression"
- Why "classification"? Just as we classify an e-mail message as spam, important, or normal, we can classify an underlying form /da/ as having the input [da], [a], [ta], etc.

probability of choosing candidate *x*

$$P(x) = \frac{e^{-\sum_{i=1}^{N} w_i C_i(x)}}{Z}$$

for all *N* constraints, sum of constraint's weight * how many times candidate *x* violates that constraint

sum of these numerators for all the candidates

[1] some people don't like to call it "OT" because it doesn't involve strict domination

	ONSET weight: 50	*VOICEDOBS weight: 19.9	MAX-C weight: 23.7	NOCODA weight: 16.4	*ɛ# weight: 4.5
☞ *a* /da/ → da		*			
b /da/ → a	*		*		
c /lob/ → lob		*		*	
☞ *d* /lob/ → lo			*		
☞ *e* /tɛf/ → tɛf				*	
f /tɛf/ → tɛ			*		*
g /kɛ/ → kɛʔ				*	
☞ *h* /kɛ/ → kɛ					*

$$P(tEf) = \frac{e^{-(50*0+19.9*0+23.7*0+16.4*1+4.5*0)}}{e^{-(50*0+19.9*0+23.7*0+16.4*1+4.5*0)} + e^{-(50*0+19.9*0+23.7*1+16.4*0+4.5*1)}} = 0.9999925$$

How are the weights chosen?

- They're chosen so that predicted probabilities for the correct outputs are as large as possible
- More precisely, maximize the sum of the logs of the predicted probabilities of the *M* pieces of data: $\sum_{i=1}^{N} \ln P(x_i)$
- We'll modify this below with a smoothing term.

How are the weights learned?

- OTSoft (and other software) will do it for you, using the Conjugate Gradient Algorithm (see Shewchuk 1994 for tutorial), a fancy version of rolling downhill.

What about free variation?

- Suppose /da/ occurs 10 times, 90% [da], 10% [a].
- If we have weights that produce 99% [da], sum of log probabilities is ln(.99+.99+.99+.99+.99+.99+.99+.99+.99+.01) = -4.696
- But if we have weights that produce 90% [da] (matching the rate in the data), sum of log probabilities is ln(.90+.90+.90+.90+.90+.90+.90+.90+.90+.10) = -3.251, which is bigger.

Why "maximum entropy"?

- The **entropy** *H* of a random variable *X* is

$$H(X) = \sum_{i=1}^{n} p(x_i) \ln(\frac{1}{p(x_i)})$$ [2]

  - That is, for every value that *X* can take on (e.g., *regular, vowel, zero*, etc.)...
  - ...multiply the probably of *X* taking that value by the log (any base) of 1 divided by that probability.
- Entropy is a measure of how unpredictable the probability distribution for *X* is.
  - Let's consider a couple of different distributions on the board and see how their entropies come out.

[2] I'm using a base-*e* log here, but the base can be anything. Often in computer science it's 2.

4 Smoothing, also known as regularization

- In the regression models we made earlier in the course, we asked the computer to find the coefficients that best fit the data.
- But we also worried about overfitting.
- One response to overfitting is to do some model comparison to decide if some independent variables should be removed.
- But another response is to (decide how much to) **penalize** coefficients that are large.
  - We want to trade coefficient size off against fit: in order to have a large coefficient, an independent variable must do a lot of work in explaining the data.

5 Smoothing in linear regression

- Previously, we asked the computer to minimize this measure of error: $\sum_{i=1}^{n}(predicted\_value\_for\_x_i - actual\_value\_y_i)^2$
  - That is, for each of the *n* data points, take the difference between its actual *y* value and the *y* value that the model predicts, and square it.
  - Minimize the sum of those squares.
- Here's how to smooth it—minimize this measure instead: $\sum_{i=1}^{n}(predicted\_value\_for\_x_i - actual\_value\_y_i)^2 + \lambda\sum_{j=1}^{m}(coefficient_m)^2$
  - That is, for each of the *m* coefficients in the model, square it, sum up those squares, and multiply by a constant λ.
  - What happens if we choose a very small λ? A very big λ?

6 Smoothing in MaxEnt

- Here was our first approximation: just maximize how probable the observed data would be under the current model: $\sum_{i=1}^{N}\ln P(x_i)$
- Second approximation: maximize that probability, *minus* a penalty for big weights: $\sum_{i=1}^{N}\ln P(x_i) - \lambda\sum_{j}^{M}w_j^2$
- Third approximation: what if it's not *big* weights we want to penalize, but weights that are different from whatever the default is for that weight? We can give each of the *M* constraints c_j its own default weight, μ_j, and penalize departures from that weight: $\sum_{i=1}^{N}\ln P(x_i) - \lambda\sum_{j}^{M}(w_j - \mu_j)^2$
- And finally, instead of just one λ, we can give each constraint c_j its own "willingness" to depart from μ_j, σ_j : $\sum_{i=1}^{N}\ln P(x_i) - \sum_{j}^{M}\frac{(w_j - \mu_j)^2}{2\sigma^2}$

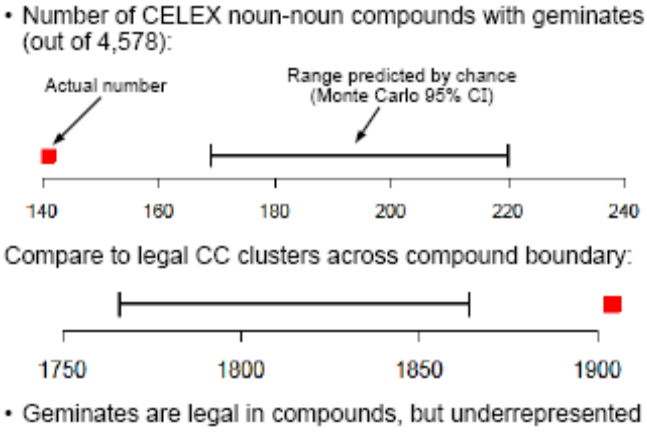
7 Do humans smooth?

- We saw that smoothing reduces overfitting, which tends to produce a better fit on future data, so that's a good reason to use it.
- As you'll experiment with in the lab, it's also useful for getting an idea of how worthwhile a constraint is.
- But do human language learners smooth? Let's look at some case studies

8 Martin 2007a,b, 2011

Facts to be accounted for

- English does not allow geminates (long/double consonants) within a morpheme: there can be no minimal pair [hæpi]/[hæppi].
- English does allow geminates in compounds and affixed words: *no[nn]egotiable*, *sou[ll]ess*, *boo[kk]ase*.
- Martin discovered, however, that geminates are less common than would be expected by chance—that is, there are not as many words like *bookcase* as expected:



(Martin 2007b)

Martin discovered similar compound underrepresentation for sibilant harmony in Navajo and vowel harmony in Turkish.

Martin's approach

- It's easy to construct a learner that can learn these facts.
- What Martin set out to do was construct a learner that, presented with no bias in compounds, will learn a bias anyway.

Martin's toy language—contains only two sounds

- The training data consists of biconsonantal clusters of [p] and [t], with an optional morpheme boundary:

Cluster	Structure	Number of examples
pt	monomorpheme	2000
tp	monomorpheme	2000
p+t	compound	1000
t+p	compound	1000
p+p	compound	1000
t+t	compound	1000

No bias in training data

- Tautomorphemic geminates [pp], [tt] do not occur in training data, but heteromorphemic geminates occur freely

(Martin 2007b)

Constraints available to learner

Structure-sensitive constraints:

*pp	no geminates within morpheme
*tp	no non-geminate clusters within morpheme
*p+p	no geminates across morpheme boundary
*t+p	no non-geminate clusters across morpheme boundary

Structure-blind constraints:

*p(+)p	no geminates
*t(+)p	no non-geminate clusters

(Martin 2007b)

Grammar

	*pp weight = 4.01	*tp weight=0.13	*p(+)p weight=.03	*t(+)p weight=.00	*p+p weight=.00	*t+p weight=.00	score
a pp	*		*				$e^{-4.04}$=0.02
☞ *b* tp		*		*			$e^{-0.13}$=0.87
c p+p			*		*		$e^{-0.04}$=0.96
d t+p				*		*	$e^{-0.00}$=1.00

- *pp* gets a low score, as expected—because *pp has a big weight
- *tp* gets a high score, as expected—because *tp has a small weight
- *t+p* gets a high score, as expected
- but *p+p* gets a slightly lower score—**because *p(+)p has a non-negligible weight**

Why does *p(+)p get non-zero weight?

- The **smoothing term** uses $(w\text{-}0)^2 = w^2$
  - So, it's better to account for data like the absence of *pp* by spreading the responsibility over two constraints—*pp and *p(+)p—than by loading all the blame onto one constraint. (Let's try the math)
- Thus, if there are structure-blind constraints like *p(+)p, generalizations that are true of one type of word (here, monomorphemes) will "leak" onto other types of word (here, compounds).

9 Wilson 2006: making the smoothing term do even more work

Velar palatalization

- Cross-linguistically, it's common for /k/ and /g/ to become [tʃ] and [dʒ] before [i]
  - and to a lesser extent before [e] and other "front" vowels (these examples, from Guion 1996, are of diachronic sound change):

(1) k > tʃ / __ {j, ɨ, i, e, ɛ, ẽ}

Pre-Proto-Slavic	OCS	Gloss
*wilk-e	vɫitʃe	'wolf' (voc.)
*pla:k-j-o:-m	platʃõ	'I cry'

(Guion ch. 2 p. 4)

(9) k, k', x > tʃ, tʃ', ʃ /__ i [5]

Proto-Salish	Cowlitz Salish	Gloss
*k'ilk	tʃ'ilk	'window'
*kitaq-	tʃæq-	'argue'
*túɬxils-	túɬʃils-	'hint'

(Guion ch. 2 p. 12)

Confusability

- This is presumably because the "fronter" articulation of /k/ and /g/ before [i,e] creates a sound that is hard to distinguish from [tʃ]/[dʒ], as can be seen for [i] in this confusability table from Guion 1998:

Confusions of velars and palatoalveolars

	Response							
Stimulus	[ki]	[t͡ʃi]	[gi]	[d͡ʒi]	[kɑ]	[t͡ʃɑ]	[gɑ]	[d͡ʒɑ]
[ki]	43	35	10	12				
[t͡ʃi]	10	85	0	5				
[gi]	4	4	71	21				
[d͡ʒi]	9	28	12	51				
[kɑ]					84	13	3	0
[t͡ʃɑ]					10	87	0	3
[gɑ]					4	0	87	9
[d͡ʒɑ]					2	23	10	65

Note. From "The Role of Perception in the Sound Change of Velar Palatalization," by S. G. Guion, 1998, Phonetica, 55, pp. 18–52. Copyright 1998 by S. Karger AG, Basel. Adapted with permission.

Wilson p. 949

(English-speaking subjects, stimuli masked by noise.)

Bias: [k,g] should be more confusable before [i] than [e], and more before [e] than [a]

Wilson devises a measure of similarity based mainly on peak spectral frequency, fitted to Guion's confusion data that would predict intermediate status for [e]:

Maximum likelihood estimates of perceptual similarities in three vowel contexts

[ki]/[t͡ʃi]	[ke]/[t͡ʃe]	[kɑ]/[t͡ʃɑ]	[gi]/[d͡ʒi]	[ge]/[d͡ʒe]	[gɑ]/[d͡ʒɑ]
9.23^{-1}	*12.68^{-1}*	88.72^{-1}	21.13^{-1}	*40.60^{-1}*	126.93^{-1}

Note. *i/j* denotes $b_j\eta_{ij}$. Values in italics are interpolated.

(Wilson p. 954)

Wilson's artificial-language-learning experiment

- Subjects in the "High" group were taught palatalization only before [i]—Wilson predicts that they won't generalize to [e], and they didn't.
- Subjects in the "Mid" group were taught palatalization only before [e]—as predicted, they generalize that the rule applies everywhere equally.

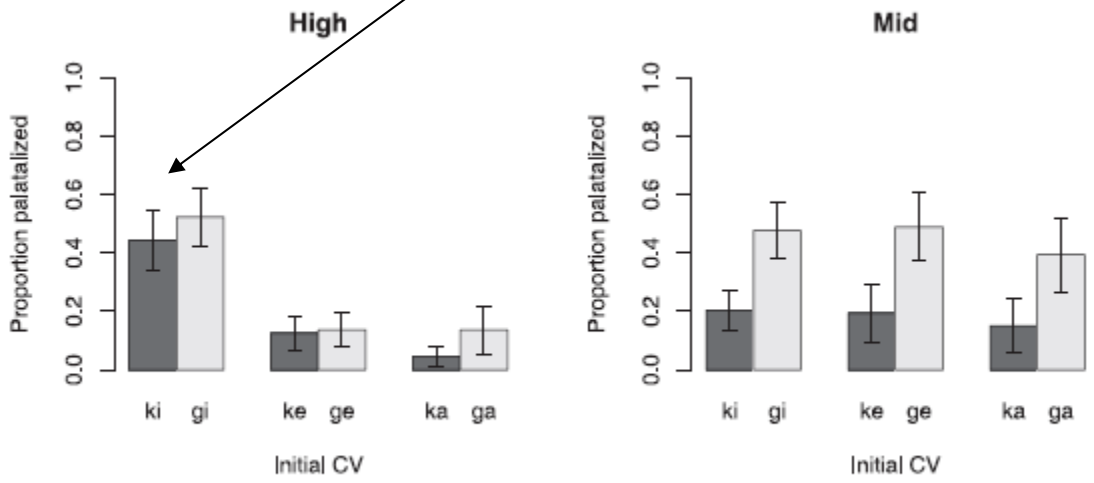


Fig. 2. Results of Experiment 1 by condition. *Note.* Error bars represent standard error of the mean.

(Wilson p. 966)

The learner

- Wilson uses the similarity values derived above to assign to each of the markedness constraints below **its own σ^2**,[3] which determines how reluctant that constraint is to move from its default weight μ (0 in all cases here).

Markedness constraints on palatalization

	Prior Values			
	Biased		Unbiased	
Constraint	μ	σ^2	μ	σ^2
*ki	0.0	9.23^{-2}	0.0	10^{-2}
*ke	0.0	12.68^{-2}	0.0	10^{-2}
*kɑ	0.0	88.72^{-2}	0.0	10^{-2}
*$kV_{[-low]}$	0.0	12.68^{-2}	0.0	10^{-2}
*$kV_{[-high]}$	0.0	88.72^{-2}	0.0	10^{-2}
*kV	0.0	88.72^{-2}	0.0	10^{-2}
*gi	0.0	21.13^{-2}	0.0	10^{-2}
*ge	0.0	40.60^{-2}	0.0	10^{-2}
*gɑ	0.0	126.93^{-2}	0.0	10^{-2}
*$gV_{[-low]}$	0.0	40.60^{-2}	0.0	10^{-2}
*$gV_{[-high]}$	0.0	126.93^{-2}	0.0	10^{-2}
*gV	0.0	126.93^{-2}	0.0	10^{-2}

- For example, in the biased learner (on left) it requires little data to increase the weight of *ki (big σ^2), but much more data to increase the weight of *kɑ.
- Also faithfulness constraints, one against changing /k/ and one against changing /g/.

Results

- For the High condition, where the subjects are essentially repeating back what they were taught, the learner does OK at matching the experimental results with or without bias.
- But for the Mid condition, the learner matches the experimental results much more closely with bias:

Correlations (r) between observed and predicted rates of palatalization in Experiment 1

Condition	Model	All Items	Critical Items
High	Substantively biased	.910 (.83)	.870 (.76)
	Unbiased	.913 (.83)	.871 (.76)
Mid	Substantively biased	.859 (.74)	.758 (.58)
	Unbiased	.550 (.30)	.396 (.16)

Note. Values in parentheses are percentage variance explained (r^2).

Wilson p. 968

10 Another interesting case: affix order in Ryan 2010

- A case where each input can have 3 or more output candidates with non-zero frequency.
- Ryan gives learner only basic data (most-frequent candidate for each input), and imposes smoothing on a noisy Harmonic Grammar by limiting the number of learning iterations.
  - Result: learner still yields a good match to the frequencies for each candidate
  - Conclusion: the speaker doesn't need to track detailed variation rates; just needs to note the main trends and be conservative (smoothing)

[3] "the prior σ of a Markedness constraint is equal to the perceptual similarity of the sounds in the greatest change that is motivated by the constraint" (p. 959)

11 Summary of the constraint models we've seen this week

name	***type of theory***	***learning algorithm***	***software for learning algorithm***
partial ordering (Anttila)	probability distribution over classic OT grammars	none, as far as I know	*NA*
Stochastic OT (Boersma)	prob. dist. over classic OT grammars	Gradual Learning Algorithm (esp. with asymmetrical plasticity—see Magri)	OTSoft, Praat
Noisy Harmonic Grammar (Boersma & Pater)	prob. dist. over Harmonic Grammars	GLA	Praat
MaxEnt (Goldwater & Johnson)		various hill-climbing methods	OTSoft, Praat, MaxEnt grammar tool

12 Presentations on Tuesday

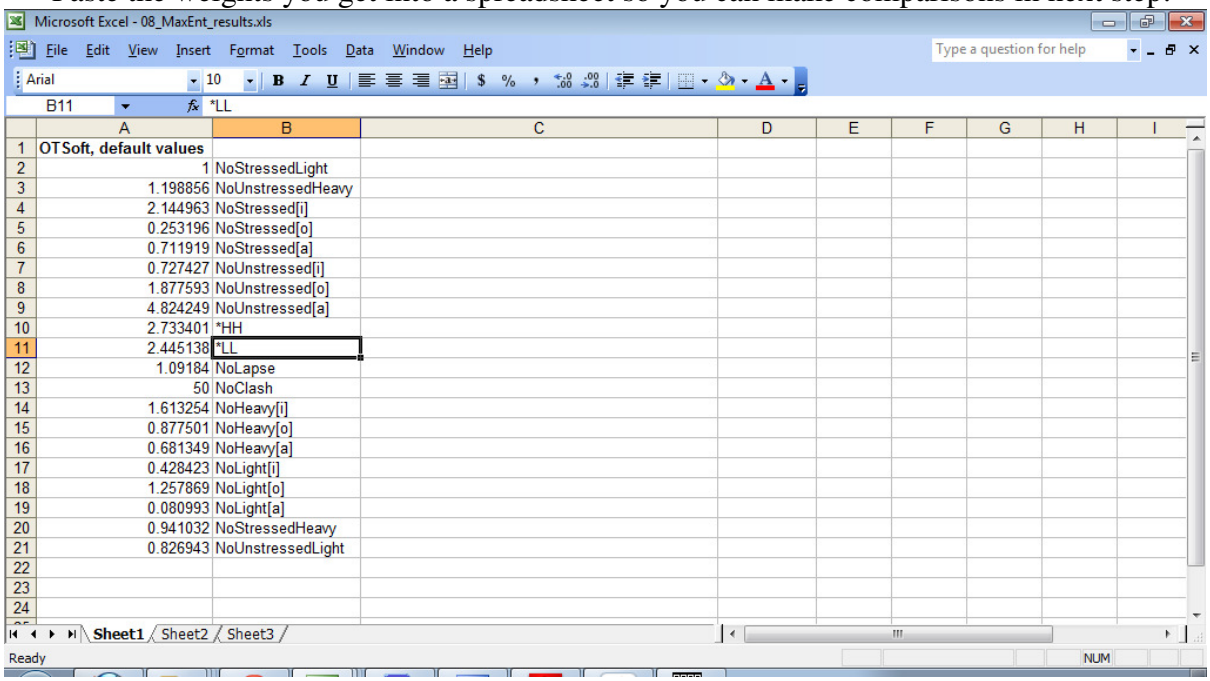
- 10 minutes each (I realized that's all we really have time for)
- This is not much time! You will have to be brief—I recommend practicing to make sure you can really do it in just 10 minutes.
- Bring a handout
- What to cover
  - Explain the phenomenon to us
  - What kind of variation is it? (free, lexical... is there multi-site variation?)
  - Where do the data come from? Or, if you don't have them yet, how will you get them?
  - Have you tried a model where you were happy with the results? If so, show us
  - Or, you could compare your values from 2 models

Lab on MaxEnt

13 MaxEnt grammar for Finnish in OTSoft

- Open OTSoft
- `Work with different file...`: choose the OTSoft input for Anttila's Finnish data.
- Choose `Maximum Entropy` and click the `Rank` button
- A new window appears. There are not many options. You can't choose μ or σ, for example.
- Click `Run` , then when it's done, `View results`.

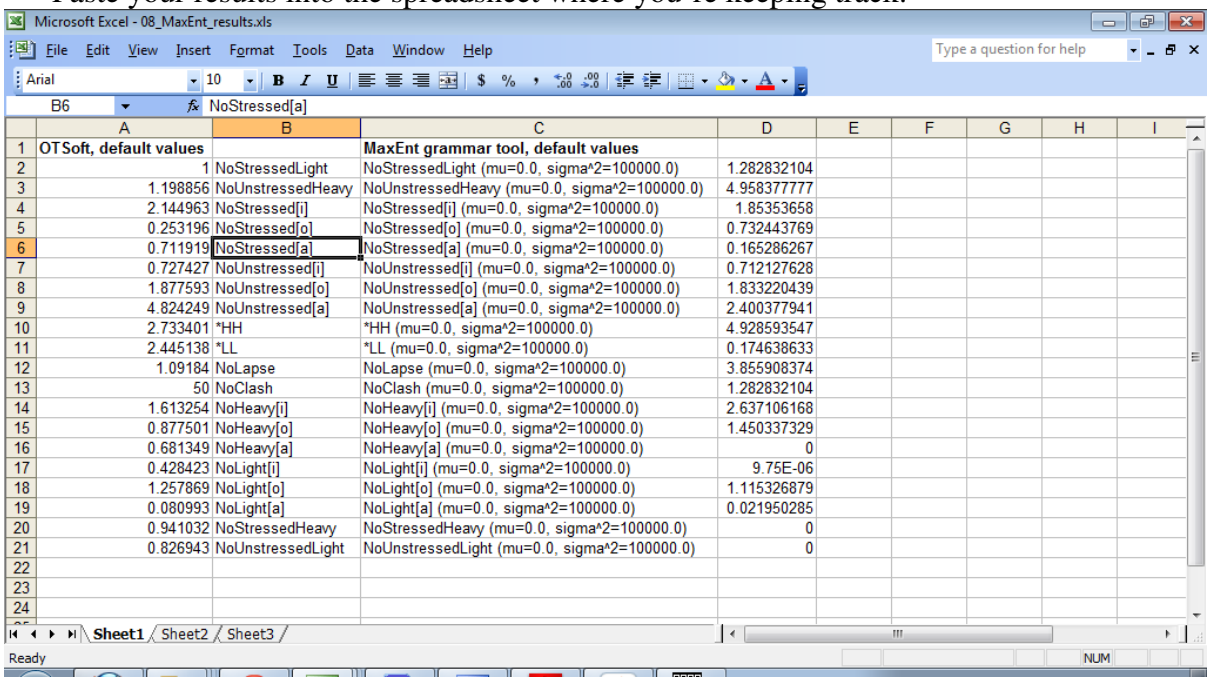
- Paste the weights you get into a spreadsheet so you can make comparisons in next step:



	A	B
1	**OTSoft, default values**	
2	1	NoStressedLight
3	1.198856	NoUnstressedHeavy
4	2.144963	NoStressed[i]
5	0.253196	NoStressed[o]
6	0.711919	NoStressed[a]
7	0.727427	NoUnstressed[i]
8	1.877593	NoUnstressed[o]
9	4.824249	NoUnstressed[a]
10	2.733401	*HH
11	2.445138	*LL
12	1.09184	NoLapse
13	50	NoClash
14	1.613254	NoHeavy[i]
15	0.877501	NoHeavy[o]
16	0.681349	NoHeavy[a]
17	0.428423	NoLight[i]
18	1.257869	NoLight[o]
19	0.080993	NoLight[a]
20	0.941032	NoStressedHeavy
21	0.826943	NoUnstressedLight

14 Finnish with the MaxEnt Grammar Tool

- From Bruce Hayes's webpage, download and unzip the MaxEnt grammar tool (http://www.linguistics.ucla.edu/people/hayes/MaxentGrammarTool/)
- Save the Finnish OTSoft input file as a *.txt file.
- Open the MaxEnt grammar tool
- Click `open tableaux` button and choose your Anttila *.txt file
- Click the `select output file` button and give a name to the file where the results will go
- Click the `Learn and report` button—so far, you haven't changed mu or sigma from the defaults
- Go and open the results file
- Paste your results into the spreadsheet where you're keeping track:

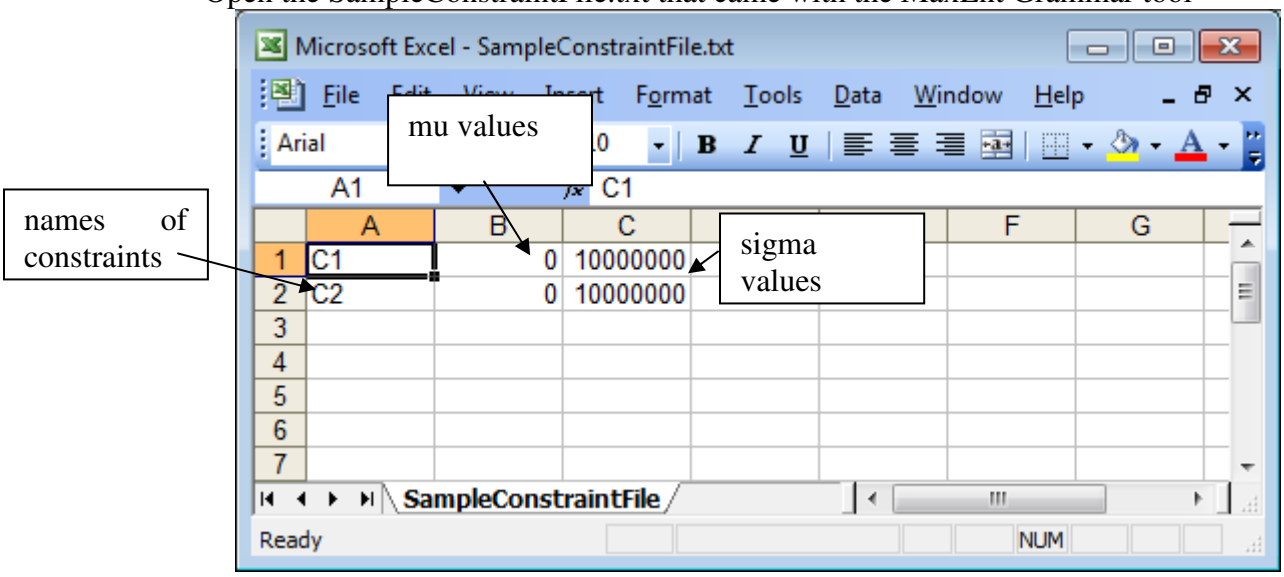


	A	B	C	D
1	**OTSoft, default values**		**MaxEnt grammar tool, default values**	
2	1	NoStressedLight	NoStressedLight (mu=0.0, sigma^2=100000.0)	1.282832104
3	1.198856	NoUnstressedHeavy	NoUnstressedHeavy (mu=0.0, sigma^2=100000.0)	4.958377777
4	2.144963	NoStressed[i]	NoStressed[i] (mu=0.0, sigma^2=100000.0)	1.85353658
5	0.253196	NoStressed[o]	NoStressed[o] (mu=0.0, sigma^2=100000.0)	0.732443769
6	0.711919	NoStressed[a]	NoStressed[a] (mu=0.0, sigma^2=100000.0)	0.165286267
7	0.727427	NoUnstressed[i]	NoUnstressed[i] (mu=0.0, sigma^2=100000.0)	0.712127628
8	1.877593	NoUnstressed[o]	NoUnstressed[o] (mu=0.0, sigma^2=100000.0)	1.833220439
9	4.824249	NoUnstressed[a]	NoUnstressed[a] (mu=0.0, sigma^2=100000.0)	2.400377941
10	2.733401	*HH	*HH (mu=0.0, sigma^2=100000.0)	4.928593547
11	2.445138	*LL	*LL (mu=0.0, sigma^2=100000.0)	0.174638633
12	1.09184	NoLapse	NoLapse (mu=0.0, sigma^2=100000.0)	3.855908374
13	50	NoClash	NoClash (mu=0.0, sigma^2=100000.0)	1.282832104
14	1.613254	NoHeavy[i]	NoHeavy[i] (mu=0.0, sigma^2=100000.0)	2.637106168
15	0.877501	NoHeavy[o]	NoHeavy[o] (mu=0.0, sigma^2=100000.0)	1.450337329
16	0.681349	NoHeavy[a]	NoHeavy[a] (mu=0.0, sigma^2=100000.0)	0
17	0.428423	NoLight[i]	NoLight[i] (mu=0.0, sigma^2=100000.0)	9.75E-06
18	1.257869	NoLight[o]	NoLight[o] (mu=0.0, sigma^2=100000.0)	1.115326879
19	0.080993	NoLight[a]	NoLight[a] (mu=0.0, sigma^2=100000.0)	0.021950285
20	0.941032	NoStressedHeavy	NoStressedHeavy (mu=0.0, sigma^2=100000.0)	0
21	0.826943	NoUnstressedLight	NoUnstressedLight (mu=0.0, sigma^2=100000.0)	0

Now try a smaller sigma. How to do this:

You need to make a "constraint data" file.

- Open the SampleConstraintFile.txt that came with the MaxEnt Grammar tool



- Using this as a model, make a file for Finnish.
- Keep the mu values at 0, but make sigma much, smaller, such as 0.01 for all the constraints
- Once you have this file, back in the MaxEnt Grammar Tool, click the `open constraints file`.
- Choose your constraints file
- Use the `select output file` button to give a new name to your next output file
- Run the learner again, and look at the results again. How do the weights and fits change?
- Play with different values of sigma

- **Optional**: Now try adding some more constraints to the Finnish OTSoft input file: perhaps you'd like to split the *V constraints according to first vs. non-first syllable.
- Remember to save the OTSoft file as *.txt
- Remember to add the new constraints to your Constraints file
- When you use a very high sigma, does this constraint get any weight? How about with a very low sigma (where each constraint really has to justify itself)?

15 Your own data

- Use the MaxEnt grammar tool to learn a MaxEnt grammar for your own data.
- Try different values of sigma
- With lower sigma, the fit will always get worse—but is this good (avoiding overfitting) or bad (underfitting)?
  - If you finished the step in Tuesday's lab where you made 10 different training and testing files, you can use them to do 10-fold cross-validation.
  - Do testing and training on all 10 files with a higher value of sigma and with a lower value
  - Which one provides a better average fit to the testing files?

16 Replicating Martin 2007

- If you still have time, make an OTSoft input file for Martin's schematic geminates (see the table in this handout).
- Remember to save it as *.txt
- Apply learning in the MaxEnt Grammar Tool under some different values of sigma.
- How much of Martin's "leakage" do you get under different values of sigma?

Once again, save all your output files!! They will be useful for your presentation.

Next time: Grammar architecture and variation patterns.

Reference

Goldwater, Sharon & Mark Johnson (2003). Learning OT constraint rankings using a Maximum Entropy model. In Jennifer Spenador, Anders Eriksson & Östen Dahl (eds.) *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*. Stockholm: Stockholm University. 111–120.

Guion, Susan (1996). Velar palatalization: coarticulation, perception, and sound change. University of Texas at Austin dissertation.

Guion, S. G. (1998). The role of perception in the sound change of velar palatalization. *Phonetica* 55: 18–52.

Guy, Gregory R. 1991. Explanation in Variable Phonology: An Exponential Model of Morphological Constraints. *Language Variation and Change* 3: 1–22.

Guy, Gregory R. 1991. Contextual conditioning in variable lexical phonology. *Language Variation and Change* 3: 223-239.

Lieberman, Erez, Jean-Baptiste Michel, Joe Jackson, Tina Tang & Martin A. Nowak. 2007. Quantifying the evolutionary dynamics of language. *Nature* 449: 713-716.

Martin, Andy. 2007a. The evolving lexicon. UCLA dissertation.

Martin, Andy. 2007b. Grammars leak: how categorical phonotactics can cause gradient phonotactics. Poster at the Workshop on Variation, Gradience and Frequency in Phonology, Stanford University.

Martin, Andy. 2011. Grammars leak: Modeling how phonotactic generalizations interact within the grammar. *Language* 87: 751-770

Ryan, Kevin M. 2010. Variable affix order: grammar and learning. *Language* 86. 758–791.

Venables, W. N. & B. D. Ripley. 2002. *Modern applied statistics with S*. 4th ed. New York: Springer.

Wilson, Colin. 2006. Learning phonology with substantive bias: an experimental and computational study of velar palatalization. *Cognitive Science* 30: 945–982.