Linguistics 251 Spring 2013
Variation in Phonology Hayes/Zuraw

Class 6, 4/18/13: More on relating maxent to logistic regression; two empirical examples

1. Assignments etc.

- Hand back Goldwater/Johnson summaries
- Hand in exercise in maxent/GLA
- Do Martin reading (on course website)
- Think about a term project

2. Where are we in the course?

- Exploring the formal side:
  - Grammar frameworks
  - Learning algorithms
- The field of statistics emerges as a kind of *doppelganger* of the field of constraint-based learning — the child, like the researcher, must find out if the pattern she observes are strong enough to be meaningful.

3. Today

- Translating between maxent and simple logistic regression.
- An application of logistic regression to a problem also attacked in maxent
- Biases: theory as a source of a new hypothesis about bias

TRANSLATING A MAXENT PROBLEM INTO LOGISTIC REGRESSION

4. Goals

- Examine an empirical domain where maxent analysis might be useful.
- Do the same analysis in maxent/logistic regression and see what similarities/differences there are.

5. McPherson and Hayes (in progress)

- We’re interested in account for a pattern whereby the three vowel harmony processes of Tommo So “peter out” as you move to the outer layers of the morphological structure.
- “Layers”: this is traditional Lexical Phonology and Morphology (Kiparsky, others)
  - Look at the order with which affixes are added to the stem, and assign them to levels accordingly. English: *class-ifie-d*, never **class-ed-ify*

6. The vowels of Tommo So, and a feature chart

a.

```
i          (ɨ)          u
    e              o
       ɛ       ɔ
           a
```

b.

	[high]	[low]	[back]	[ATR]
i	+	−	−	0
e	−	−	−	+
ɛ	−	−	−	−
a	−	+	0	0
ɔ	−	−	+	−
o	−	−	+	+
u	+	−	+	0
ɨ	+	−	0	0

- [ɨ] is a non-phonemic reduced vowel, occurring only in medial syllables.

7. The three vowel harmony processes of Tommo So

- Low harmony
- Backness harmony
- ATR (Advanced Tongue Root) harmony

8. Low harmony

"A non-high vowel takes on the same value of [low] as the initial vowel."

In rule notation: $\begin{bmatrix}\text{V}\\ -\text{high}\end{bmatrix} \rightarrow [\alpha\text{low}] \mathbin{/} \#\, C_0 \begin{bmatrix}\text{V}\\ \alpha\text{low}\end{bmatrix} X \_\_\_$

a.	/dʒàá-ndɛ́/ →	[dʒàà-ndá]	'meal-FACTITIVE' = 'cook'
b.	/pándá-ílɛ́/ →	[pánd-ílá]	'widow-REVERSIVE' = 'marry a widow'

9. Backness harmony

"A vowel takes on the same value of backness (and rounding) as the initial vowel."

$$\text{V} \rightarrow \begin{bmatrix}\text{V}\\ \alpha\text{back}\\ \alpha\text{round}\end{bmatrix} \mathbin{/} \#\, C_0 \begin{bmatrix}\text{V}\\ \alpha\text{back}\end{bmatrix} X \_\_\_$$

a.	/ɲ́jɛ́-mɔ́/ →	[ɲ́jɛ́-mɛ́]	'eat-CAUSATIVE'
b.	/tɔ́bɔ́-íjɛ́/ →	[tɔ́b-ɨ́jɔ́]	'turban-MEDIOPASSIVE' = 'put on a turban'

10. ATR harmony

"A mid vowel takes on the ATR value of a preceding mid vowel."

$$\begin{bmatrix}\text{V}\\ -\text{high}\\ -\text{low}\end{bmatrix} \rightarrow [\alpha\text{ATR}]\ /\ \begin{bmatrix}\text{V}\\ -\text{high}\\ -\text{low}\\ \alpha\text{ATR}\end{bmatrix} \text{X}\ \_\_\_$$

a.	/dèé-ndɛ́/	→	[dèè-ndé]	‘know-FACTITIVE’ = ‘introduce’
b.	/gòró-íjɛ́/	→	[gòr-ɨ́jó]	‘hat-MEDIOPASSIVE’ = ‘put on a hat’

- The second form is an example of both backness and ATR harmony.

11. Vowel harmony in stems

- Tommo So stems are sharply restricted in their possible vowel combinations.
- Most gaps can be explained by assuming that harmony (above) applies stem-internally.
    - ➢ ■ = absent due to ATR harmony; e.g. *e ɛ
    - ➢ ▒ = backness harmony; e.g. *e o
    - ➢ ▥ = low harmony; e.g. *a ɔ
- Additional gap: □ Verb stems may not end in a high vowel (/i/ and /u/ columns).[1]

				Second vowel				
		i	e	ɛ	a	ɔ	o	u
First vowel	i	9	39	56	▥	▒	▒	▒
	e		37	■	▥	■	▒	▒
	ɛ	4	■	79	▥	▒	■	▒
	a	2	▥	▥ 4	151	▥	▥	4
	ɔ	▒	■	▒ 2	▥	100	■	2
	o	▒	▒	■	▥	■	46	4
	u	▒	▒	▒ 6	▥	43	43	8

12. Suffix harmony again

- The three harmony processes apply to six different suffixes, deriving the following surface allomorphs:

Suffix	*UR*	*Surface forms*
Factitive	/-ndɛ/	[-nde, -ndɛ, -nda, -ndɔ, -ndo]
Reversive	/-ilɛ/	[-ile, -ilɛ, -ila, -ilɔ, -ilo]
Transitive	/-irɛ/	[-ire, -irɛ, -irɔ, -iro]
Mediopassive	/-ijɛ/	[-ije, -ijɛ, -ijɔ, -ijo]

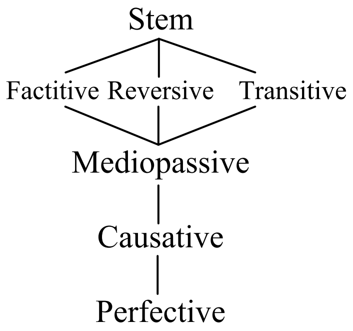
[1] Attested numbers in the table are the result of medial vowel reduction (e.g. *ádúbá* ‘think’).

Causative	/-mɔ/	[-mɛ, -mɔ]
Defocalized perfective	/-i/	[-i, -u]

- Why are some allomorphs missing? see below

13. Level ordering in Tommo So

- To justify a level-ordered system, we inspected the data for all cases of “stacked” affixes—multiple affixes attached in sequence to the same stem.
    - e.g. *àmà-nd-ìjɛ̀-m-ì* ‘rancid-factitive-mediopassive-causative-perfective’, ‘he made it rancid’
- Ordering was completely consistent; no pairs with opposite affix order.
- All observed precedence relations, plotted as Hasse diagram:



14. Our posited levels

- We assume not five levels but seven.

1.	Stem		
2.	Factitive	/-ndɛ/	(derivation)
3.	Reversive	/-ilɛ/	(derivation)
4.	Transitive	/-irɛ/	(derivation)
5.	Mediopassive	/-ijɛ/	(derivation)
6.	Causative	/-mɔ/	(derivation)
7.	Defocalized perfective	/-i/	(inflection)

- Placing Transitive in a level “outside” Factitive and Reversive is plausible — it’s more productive and semantically transparent than the latter two.
- The Factitive/Reversive level difference is stipulated, to get the harmony facts (below).
- Our basic conclusions would hold under a coarser level system.

15. Harmony rates decrease as one moves to outer levels

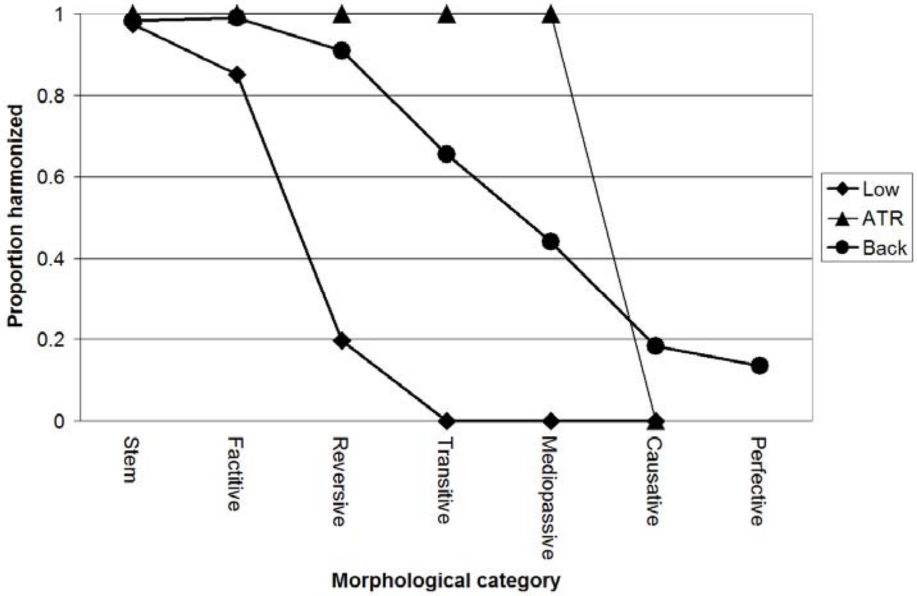
- Frequency data: based on a corpus of 2193 examples compiled from McPherson’s field notes and recorded texts.

- We used token frequency, since all words vary alike (not word-by-word variation).
- Harmony rates by level:

	Low	*%*	*Backness*	*%*	*ATR*	*%*
1. Stem[2]	151/155	97.4	470/478	98.3	262/262	100
2. Factitive	57/67	85.1	95/96	99	80/80	100
3. Reversive	12/61	19.7	40/44	90.9	43/43	100
4. Transitive	0/15	0	38/58	65.5	31/31	100
5. Mediopassive	0/169	0	107/143	74.8	231/231	100
6. Causative	0/42	0	13/71	18.3	0/43	0
7. Perfective	N/A	—	17/125	13.6	N/A	—

16. Plotting the proportion of tokens that undergo each harmony process

- Why missing values? Because Perfective /-i/ is only eligible for Backness harmony.



17. Our idea

- Closeness to the stem is a *number*; bigger for earlier morphological levels.

Stem	>>	Factitive	>>	Reversive	>>	Transitive	>>	Mediopassive	>>	Causative	>>	Perfective
7	>>	6	>>	5	>>	4	>>	3	>>	2	>>	1

- Violations for AGREE() constraints are multiplied by the stem-closeness value — another knob.
- IDENT() is straightforward; one IDENT() constraint for each harmonizing feature.

[2] Stem rates calculated as follows: denominator = number of forms that match the structural description of the rules in (8)-(10); we make no claims about the UR's of harmonized stems.

- Here is our chart of violations. Organization:
  - morphological construction at which harmony is at stake
  - harmonizing feature
  - whether you get harmony for that feature or not, with frequencies

			Ident (low)	Ident (ATR)	Ident (back)	Agree (low)	Agree (ATR)	Agree (back)
Stem/low	harmony	151	1					
	no harmony	4				7		
Stem/ATR	harmony	262		1				
	no harmony	0					7	
Stem/back	harmony	470			1			
	no harmony	8						7
Stem+factitive/low	harmony	57	1					
	no harmony	10				6		
Stem+factitive/ATR	harmony	80		1				
	no harmony	0					6	
Stem+factitive/back	harmony	95			1			
	no harmony	1						6
Stem + Reversive/low	harmony	12	1					
	no harmony	49				5		
Stem + Reversive/ATR	harmony	43		1				
	no harmony	0					5	
Stem + Reversive/back	harmony	40			1			
	no harmony	4						5
Stem + Transitive/low	harmony	0	1					
	no harmony	15				4		
Stem + Transitive/ATR	harmony	31		1				
	no harmony	0					4	
Stem + Transitive/back	harmony	38			1			
	no harmony	20						4
Stem + Mediopassive/low	harmony	0	1					
	no harmony	169				3		
Stem + Mediopassive/ATR	harmony	231		1				
	no harmony	0					3	
Stem + Mediopassive/back	harmony	107			1			
	no harmony	136						3
Stem + Causative/low	harmony	0	1					
	no harmony	42				2		
Stem + Causative/ATR	harmony	0		1				
	no harmony	43					2	
Stem + Causative/back	harmony	20			1			
	no harmony	58						2
Stem + Perfect/low	harmony	0	1					
	no harmony	0				1		
Stem + Perfect/ATR	harmony	0		1				
	no harmony	0					1	
Stem + Perfect/back	harmony	17			1			
	no harmony	108						1

18. Pedagogical apology

- There's actually a whole set of *additional* inputs and candidates: stems where no trigger vowel is present.
- Because harmonically bounded candidates can partly-win in maxent, we have to make sure that the IDENT() constraints are weighted high enough to kill off these unwanted candidates.
    - Specifically: winner violates no constraints (no harmony needed, outcome is faithful); loser violates IDENT().

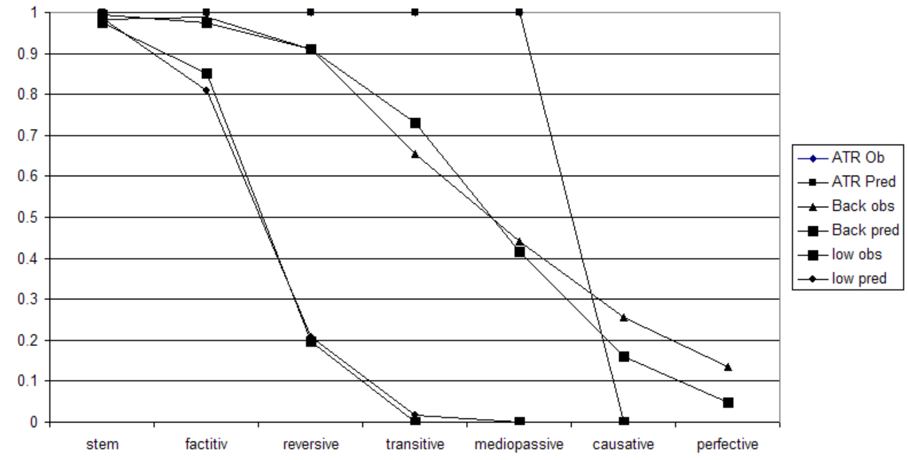
19. Running this in maxent

- Using the Maxent Grammar Tool, we get these weights:

Constraint	Weight
IDENT(LOW)	15.2
IDENT(ATR)	51.2
IDENT(BACK)	4.3
AGREE(LOW)	2.8
AGREE(ATR)	20.9
AGREE(BACK)	1.3

20. How well does it work?

Fit to data is not bad:



- What about keeping the harmonically-bounded candidates in check?
    - Highest probability assigned is .013, which is higher than we wish but not too bad.
- We don't have any significance figures for our six constraints — this is bad.

21. Sigmoids

- The Tommo So example is one of many belong to a class:
    - gradient constraint (AGREE) vs. non-gradient constraint (IDENT)
    - when you plot frequency against the scale of gradience, you get a **sigmoid** curve.
    - Nice pretty curve with level peripheries, sloping middle
    - These curves show up all over in linguistics (see McPherson/Hayes papers, and later in this course).

REDOING THE TOMMO SO MODEL WITH LOGISTIC REGRESSION

22. When is it feasible to redo a maxent problem in logistic regression?

- You need a **constant number of plausible candidates** for each input; which usually will be **two** (e.g. vowel harmony, choice of two allomorphs).
    - Background assumption: there are other constraints, with essentially infinite weight, that rule out silly candidates like deleting disharmonic vowels.
- You need the candidates to be **in parallel**.
    - Vowel harmony: harmonized/not harmonized.[3]

23. Weights in maxent and logistic regression

- Maxent
    - Applied to OT, maxent is a **linguistic theory**; it assumes that grammars consist of constraints that assigned violations.
    - Only rarely have people (e.g. Flemming 1995 et seq. dispersion theory) proposed **licenses**; i.e. function that reward particular phonological properties.
    - Why favor constraints? Example: [ta] is good causes an infinite string of [ta]'s to be appended to the stem in the winning candidate.
    - Upshot: *linguistic theory constrains the weights to be positive.*

- Logistic regression
    - Instead of constraints we have **independent variables**, which could be anything.
    - We're fine with weights being negative.

24. A possibly-convenient way to convert a maxent analysis to logistic regression

- This handout explores **violation subtraction**.
- Here, I'm subtracting the violations of the unharmonized candidate from the harmonized one:
    - sample row of an OTSoft file

[3] Note that the application here depends on Hayes and McPherson idealizing their data to this binary choice. Another thing they could have done is simply assign probability to all seven vowels of Tommo So for each suffix (on the agenda).

		freq.	Ident (low)	Ident (ATR)	Ident (back)	Agree (low)	Agree (ATR)	Agree (back)
Stem/low	harmony	151	1					
	no harmony	4				7		

- subtracted:

Stem/low	harmony	151	1			-7		
Stem/low	no harmony	4	1			-7		

- The nice part is that this lets you go with all-nonnegative weights, just like in maxent.

- I also turned the outputs into arbitrary numbers (“Winner index”), with 1 = harmony, 0 = no harmony.

		Winner Index	freq.	Ident (low)	Ident (ATR)	Ident (back)	Agree (low)	Agree (ATR)	Agree (back)
Stem/low	harmony	1	151	1			-7		
Stem/low	no harmony	0	4	1			-7		

- This is done in a version of OTSoft that is currently in beta.
- We now have a pure statistical problem:
  - find the weights in logistic regression that best predict the value WinnerIndex
  - and I’ve jiggered the problem so that the weights will be positive, like in maxent.
  - Why? Intuitively: harmony is target value 1, and is favored by IDENT constraints, with positive violations. No-harmony is target value 0, and is favored by AGREE constraints, with negative violations.

25. Swooping into R

- R: a consortium of statisticians making their own package for free
- Source: http://www.r-project.org/
- R is diverse and powerful, and is continually growing as people (e.g. Harald Baayen) invent packages.
- If you save your “scripts”, you can repeatedly do the same things to different data.
- No one thinks R is easy — it’s very unforgiving of random user error, and error is hard to diagnose.
- Kie and Robert seem to be virtuosi …

26. Some things that will help you with R

- Start with an existing script if you can.
- Input files can be tab-delimited text, which is easy to make.

- Be *ultraconservative* in naming your variables. All-letter variables is always safe.[4]
- R is case-sensitive (the hallmark of user-hostile software everywhere …)
- No empty cells: if missing values, enter "NA".

27. Bits of R code from my script

```
library(languageR)
library(arm)
```

"Grab some libraries you'll need later to run various commands" (when you learn about the command, you can usually learn about the library it's in; and the libraries are easily loaded from the Packages menu on the R interface.)

```
MyData=read.table("TommoSoForR.txt", header=T, sep="\t")
```

"Open the file TommoSoForR.txt, which has a header row, and is tab-separated. Put what you read into an object, called MyData, which may be referred to later as such."

```
colnames(MyData)
```

"On the normal output screen, tell me all the column headers in order." Thus:

```
[1] "Input"          "Winner"         "WinnerIndex"    "WhichToken"
[5] "HowManyTokens"  "Identlow"       "IdentATR"       "Identback"
[9] "Agreelow"       "AgreeATR"       "Agreeback"
```

Now the big move:

```
GrandFullModel = bayesglm(WinnerIndex ~ + Identlow + IdentATR + Identback
+ Agreelow + AgreeATR + Agreeback -1, data = MyData, family="binomial")
```

Or in easier-to-edit format:

[4] Details: http://cran.r-project.org/doc/FAQ/R-FAQ.html#What-are-valid-names_003f

```
MyModel = bayesglm(WinnerIndex ~ + +
Identlow + +
IdentATR + +
Identback + +
Agreelow + +
AgreeATR + +
Agreeback -1, data = MyData, family="binomial")
```

"Use the bayesglm function to do regression.[5]	`bayesglm()`
Try to predict as best as you can the value of WinnerIndex	`WinnerIndex; 1 for harmony, 0 for no harmony`
on the basis of	`~`
six independent variables	`Identlow + IdentATR + Identback + Agreelow + AgreeATR + Agreeback`
Do not include an intercept term[6]	`-1`[7]
The data you should fit is in the object MyData	`data = MyData,`
The particular kind of regression to be used is binomial logistic regression	`family="binomial"`

28. Results

```
summary(MyBayesModel)
```

Puts a summary on the screen:

```
Coefficients:
          Estimate Std. Error z value Pr(>|z|)
Identlow  11.17248    0.97119   11.50   <2e-16 ***
IdentATR   9.29154    0.91035   10.21   <2e-16 ***
Identback  4.27508    0.18506   23.10   <2e-16 ***
Agreelow   2.06730    0.17141   12.06   <2e-16 ***
AgreeATR   4.13129    0.35254   11.72   <2e-16 ***
Agreeback  1.31360    0.05914   22.21   <2e-16 ***
```

The last column makes us happy because every constraint is hugely significant.

[5] You can also just say "glm", General Linear Model. The Bayes version (http://www.stat.columbia.edu/~gelman/research/unpublished/priors7.pdf) is designed to be more accurate when some of your constraints prefer only winners.

[6] In this context, an Intercept term is one that penalizes all outputs of a given type. Here, depending on the weight it gets, it could be a constraint that says APPLY HARMONY or DON'T APPLY HARMONY. We're using a more nuanced system, with constraints specific to each feature.

[7] Yes, this seems to be the way you turn off the intercept term. There used to be a tag `Intercept = False`, but it wasn't obscure enough, so they took it out.

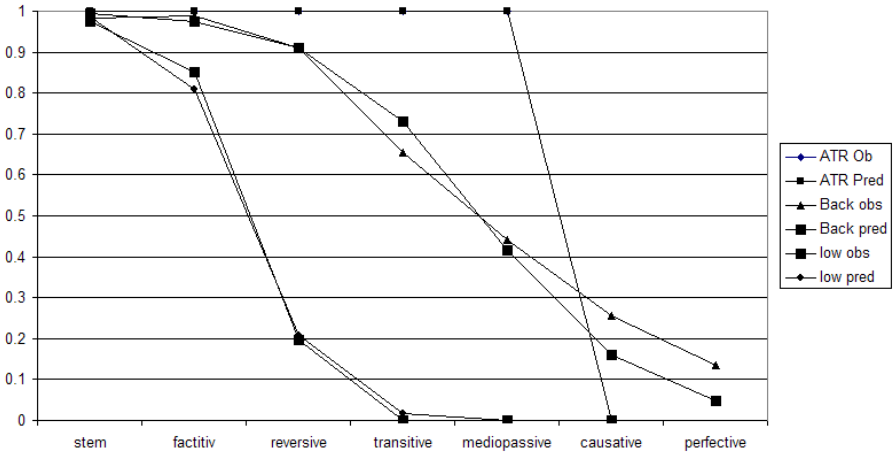
29. How did it come out? Weights of maxent vs. logistic regression

Constraint	MGT	R
Ident(low)	15.19	11.17
Agree(low)	2.77	2.07
Ident(ATR)	51.17	9.29
Agree(ATR)	20.87	4.13
Ident(back)	4.33	4.28
Agree(back)	1.33	1.31

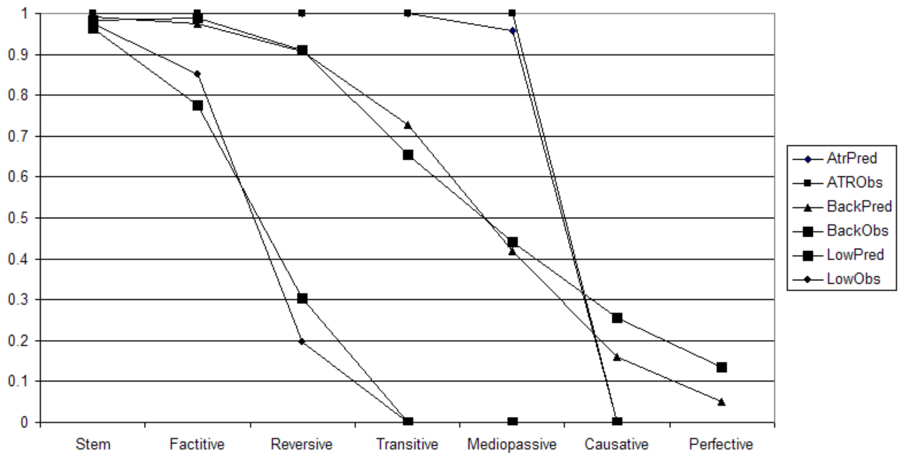
- For low harmony and Backness harmony, weights are rather similar.
- For ATR harmony, there is a very sharp cutoff, necessitating huge weights for maximaum accuracy.
    - ➢ MGR, with a very weak prior (σ = 100000) went for a close fit.
    - ➢ bayesglm() was skeptical that the world could be like this and went for much lower weights.

30. Fit to data

Maxent Grammar Tool with maxent:

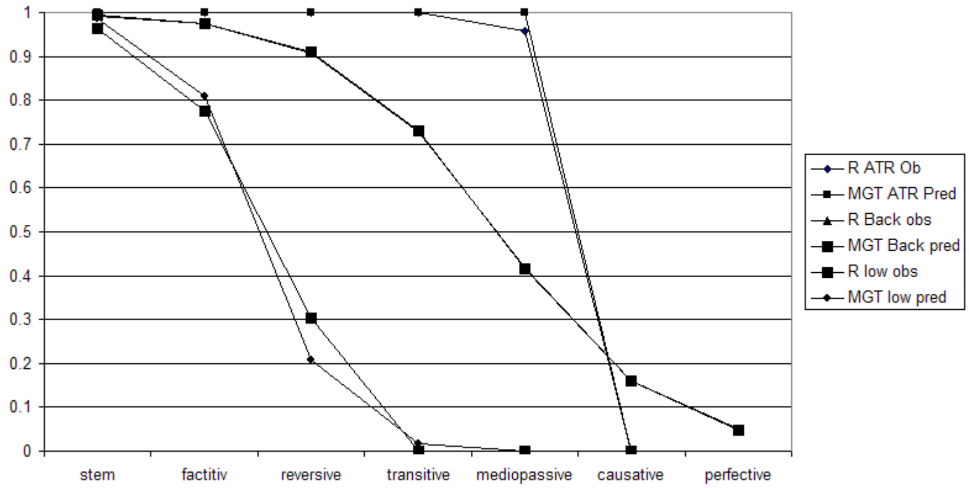


- R with bayesglm() logistic regression:



  - ➢ Note lesser fit for ATR harmony; more on this below.

- MGT vs. R:



31. Does significance testing pay off?

- Simulation: I added 20 constraints to the input file with complete random violations (1 or 0).
- Maxent Grammar Tool is not guiding us to the truth; it gives positive weights to 8 of these random constraints.
  - ➢ Largest spurious weight assigned is 1.3 — greater than AGREE(back)

Constraint	*Mu*	*Sigma*	*Weight*
Ident(low)	0	100000	9.8
Ident(ATR)	0	100000	62.3
Ident(back)	0	100000	4.2
Agree(low)	0	100000	1.9
Agree(ATR)	0	100000	26.1
Agree(back)	0	100000	1.2
Random1	0	100000	**1.3**
Random2	0	100000	0.5
Random3	0	100000	0.0
Random4	0	100000	0.0
Random5	0	100000	0.0
Random6	0	100000	0.0
Random7	0	100000	0.0
Random8	0	100000	0.0
Random9	0	100000	**0.5**
Random10	0	100000	**0.6**
Random11	0	100000	**1.0**
Random12	0	100000	**0.8**
Random13	0	100000	0.0
Random14	0	100000	0.0
Random15	0	100000	0.0
Random16	0	100000	0.0
Random17	0	100000	0.0
Random18	0	100000	**1.6**
Random19	0	100000	**0.3**
Random20	0	100000	**0.1**

- Here is what bayesglm() did:

Constraint	Estimate	Std. Error	z value	Pr(>|z|)
Identlow	10.179	1.099	9.263	0
IdentATR	6.272	1.141	5.498	0
Identback	4.802	0.71	6.759	0
Agreelow	2.057	0.228	9.017	0
AgreeATR	2.387	0.343	6.951	0
Agreeback	1.354	0.235	5.75	0
Random1	-0.679	0.519	-1.308	0.191
Random2	-1.701	0.593	-2.869	**0.004**
Random3	-0.279	0.607	-0.459	0.646
Random4	-0.046	0.53	-0.087	0.930
Random5	0.095	0.814	0.117	0.907
Random6	-1.078	0.785	-1.373	0.170
Random7	-0.109	0.612	-0.179	0.858
Random8	-1.001	0.641	-1.561	0.119
Random9	-1.158	0.561	-2.065	0.039

Random10	1.234	0.866	1.426	0.154
Random11	0.456	0.837	0.545	0.586
Random12	0.985	0.553	1.782	0.075
Random13	-0.223	0.437	-0.509	0.610
Random14	1.476	0.685	2.155	**0.031**
Random15	-2.025	0.604	-3.351	**0.001**
Random16	-0.117	0.786	-0.148	0.882
Random17	0.468	0.808	0.579	0.562
Random18	0.958	0.674	1.421	0.155
Random19	-0.095	0.581	-0.163	0.871
Random20	0.797	0.564	1.413	0.158

- This seems to be better; but given how it came out I'd still want to use a quite stringent significance criterion.
- Later, we'll consider other statistical tests and see if they do any better…

32. Final review of the math: logistic regression to maxent

- Calculate probability of the harmonized candidate.
- In logistic regression, the probability of the harmonized candidate is[8]

$$\frac{1}{1 + e^{-\text{HarmonyOfHarmonized}}}$$

- But look at the sneaky subtraction trick we did with the violations, from (24) above:

		freq.	Ident (low)	Ident (ATR)	Ident (back)	Agree (low)	Agree (ATR)	Agree (back)
Stem/low	harmony	151	1			-7		
Stem/low	no harmony	4	1			-7		

Each candidate tells of the other's violations!
So actually, HarmonyOfHarmonized — taken in the sense of a maxent grammar, is really **HarmonyOfHarmonized - HarmonyOfUnharmonized.**

So, now, in the maxent world, we have:

$$\frac{1}{1 + e^{-(\text{HarmonyOfHarmonized} - \text{HarmonyOfUnharmonized})}}$$

- Remove the outer double minus sign by switching the direction of subtraction:

[8] http://en.wikipedia.org/wiki/Logistic_regression

$$\frac{1}{1 + e^{(\text{HarmonyOfUnharmonized} - \text{HarmonyOfHarmonized})}}$$

- Trick: multiply top and bottom by $e^{-\text{HarmonyOfUnharmonized}}$:

$$\frac{e^{-\text{HarmonyOfUnharmonized}}}{e^{-\text{HarmonyOfUnharmonized}} * (1 + e^{\text{HarmonyOfUnharmonized} - \text{HarmonyOfHarmonized}})}$$

Distributing on the bottom:

$$\frac{e^{-\text{HarmonyOfUnharmonized}}}{e^{-\text{HarmonyOfUnharmonized}} + (e^{-\text{HarmonyOfUnharmonized}} * e^{\text{HarmonyOfUnharmonized} - \text{HarmonyOfHarmonized}})}$$

Multiplication is addition of exponents:

$$\frac{e^{-\text{HarmonyOfUnharmonized}}}{e^{-\text{HarmonyOfUnharmonized}} + e^{-\text{HarmonyOfUnharmonized} + \text{HarmonyOfUnharmonized} - \text{HarmonyOfHarmonized}}}$$

Adding opposites to get zero:

$$\frac{e^{-\text{HarmonyOfUnharmonized}}}{e^{-\text{HarmonyOfUnharmonized}} + e^{-\text{HarmonyOfHarmonized}}}$$

which is exactly the maxent formula.

BIASES

33. Is it good to expect weights to be small?

- When you hear, say, 5 examples of output A and 0 examples of output B, you can, in principle, set the weight of PREFERA extremely high, and get a superb fit.
    - ➢ Maxent in OTSoft goes immediately for a weight of 50, and achieves a huge number of decimal places of accuracy. Hooray!
    - ➢ Maxent Grammar Tool, in its default settings, computes weight of 10.1, and accuracy is to five decimal places: 0.99996.
- Many people would prefer the more conservative value, or even more conservative.
- Suppose instead that there are one million examples of A, 0 of B.
    - ➢ Maxent in OTSoft: same as before.
    - ➢ Maxent Grammar Tool now computes weight of 21.6; accuracy is to ten decimal places.
- So it seems sensible to want weights to be small, pending enough evidence to raise them high.

34. Bias in the Tommo So modeling

- Two different procedures got very different weights for the constraints governing ATR (see (29) above).
    - *Relative* difference is there, but bayesglm( ) was more skeptical about these weights.
- The Maxent Grammar Tool can also be made skeptical about high weights. We add to the input file a supplementary file embodying bias.

```
Constraint   μ     σ
Ident(low)   0     10
Ident(ATR)   0     10
Ident(back)  0     10
Agree(low)   0     10
Agree(ATR)   0     10
Agree(back)  0     10
```

where

μ = "preferred" value of a constraint
σ = how much the algorithm should be willing to deviate from the preferred value

- Now its weights look more like what bayesglm() did:

Constraint	Maxent Grammar Tool, sigma at default of 100000	Maxent Grammar Tool, sigma at 10	bayesglm() in R
Ident(low)	15.2	*11.1*	11.2
Agree(low)	2.8	*2.1*	2.1
Ident(ATR)	51.2	*11.8*	9.3
Agree(ATR)	20.9	*5.1*	4.1
Ident(back)	4.3	*4.3*	4.3
Agree(back)	1.3	*1.3*	1.3

35. Bias in the tiny example

- When we set μ to zero, σ to 10 for our tiny example, then
- Weight of FavorA is now just 2.3
- Predicted probability of candidate A is now just .908, which strikes me as sensible.

THE FORMAL BASIS OF SIGMA AND MU

36. Stepping back a bit first: smoothing in linear regression

- Simple linear regression: the simplest possible model that could do any good:
    - y = mx + b
    - Given a cloud of points, find the m and b that best permit you to predict y from x.

- Method of calculation: you ask the computer to minimize this 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 a typical way to smooth—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?

37. Smoothing in MaxEnt

- <u>Here was our first approximation</u>: just maximize how probable the observed data would be under the current model: $\sum_{i=1}^{N}\ln P(x_i)$
- <u>Second approximation</u>: maximize that probability, *minus* a penalty for big weights: $\sum_{i=1}^{N}\ln P(x_i) - \lambda\sum_{j}^{M} {w_j}^2$
- <u>Third approximation</u>: 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 <u>departures</u> from that weight: $\sum_{i=1}^{N}\ln P(x_i) - \lambda\sum_{j}^{M}(w_j - \mu_j)^2$
- <u>And finally</u>, instead of just one λ, we can give each constraint c_j its own "willingness" to depart from μ_j. Call it σ_j : $\sum_{i=1}^{N}\ln P(x_i) - \sum_{j}^{M}\frac{(w_j - \mu_j)^2}{2\sigma^2}$

> ➢ *BH: In the simulations above, we always used μ = 0, and played with different values of σ. We didn't try different μ, σ for different constraints, though we could have.*

38. This smoothing term is often called a Gaussian prior (and it's not the only choice!)

Why "Gaussian"?

- The equation for the normal distribution, also know as Gaussian distribution, is $y = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$ (I'll illustrate this on the board)

- Suppose we wanted to maximize: $\ln(prob(data)) + \sum_{j=1}^{M} \ln\left(\frac{1}{\sqrt{2\pi\sigma_j^{2}}} e^{\frac{-(w_j-\mu_j)^2}{2\sigma_j^{2}}} \right)$

- i.e.,maximize: $\ln(prob(data)) + \sum_{j=1}^{M}\left(\ln\left(\frac{1}{\sqrt{2\pi\sigma_j^{2}}} \right) + \ln\left(e^{\frac{-(w_j-\mu_j)^2}{2\sigma_j^{2}}} \right)\right) =$

$$\ln(prob(data)) + a\_number\_that\_doesnt\_depent\_on\_weights + \sum_{j=1}^{M} \frac{-(w_j-\mu_j)^2}{2\sigma_j^{2}}$$

- only thing learner can change is weights, so same as maximizing $\ln(prob(data)) - \sum_{j=1}^{M} \frac{(w_j-\mu_j)^2}{2\sigma_j^{2}}$

Why “prior”?

- Recall Bayes’ Law from yesterday’s seminar:

“posterior”: how probable is model given observed data

“likelihood”: how probable are data according to

“prior”: how probable is model to begin with?

$$prob(model \mid data) = \frac{prob(data \mid model) * prob(model)}{prob(data)}$$

“evidence”: usually irrelevant, since same for all models—just needed to ensure probabilities sum to 1

- Taking the log, $\overset{\text{posterior}}{\ln p(model|data)} = \overset{\text{likelihood}}{\ln p(data|model)} + \ln p(model) - \ln p(data)$

  - Compare and contrast this to our MaxEnt objective function with smoothing.

39. Coming up

- Do humans smooth? Some case studies.
- Model comparison: how do we decide which model strikes the better balance between fitting too tightly and too loosely?