

Class 5 (16 April 2013): Relation of MaxEnt to logistic regression

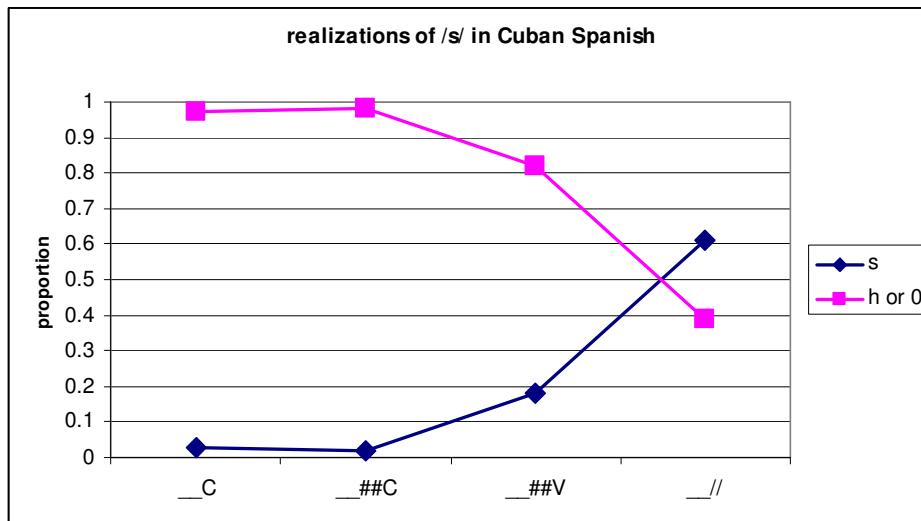
To do

- Read first “bias” paper (TBA in class)

Overview: As was spotted in the last class, MaxEnt is suspiciously similar to logistic regression. We’ll talk about logistic regression to get more solid on the math and bring out some important points like bias/smoothing.

1 MaxEnt quick review—modelling Spanish *s*-weakening

- Bybee 2001 (data from Terrell 1977; Terrell 1979; Hooper 1981): weakening of /s/ in Cuban Spanish depends on phonological context:



Since there are only 2 outcomes here, one line is predictable from the other.

- Fit a MaxEnt grammar to these input data:

			Faith	*sC	*s##C	*s##V	*s//
escuela	e[s]cuela	0.03		1			
	e[h]cuela, e[]cuela	0.97	1				
dos veces	do[s]	0.02			1		
	do[h]/do[]	0.98	1				
dos equis	do[s]	0.18				1	
	do[h]/do[]	0.82	1				
dos	do[s]	0.61					1
	do[h]/do[]	0.39	1				

- Weights learned:

0.634783 Faith
 4.110882 *sC
 4.526603 *s##C
 2.151131 *s##V
 0.187471 *s//

- Work out the predicted probability of *e[s]cuela*.

2 What if we did a logistic regression instead?

- Data to fit looks like this—simplifying assumption that there were 100 observations for each input type:

context	C	context	WdC	context	WdV	context	Final	outcome	
	1		0		0		0	faithful	e.g., e[s]cuela
	1		0		0		0	faithful	
	1		0		0		0	faithful	
	1		0		0		0	reduced	e.g., e[h]cuela
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	
	1		0		0		0	reduced	

- Resulting regression model (R Development Core Team 2010):

```
glm(formula = outcome ~ context__C + context__WdC + context__WdV +
     context__Final, family = "binomial", data = cuban_binary)
```

```
Coefficients: (1 not defined because of singularities)
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.4473     0.2050  -2.182   0.0291 *
context__C      3.9234     0.6210   6.318 2.66e-10 ***
context__WdC    4.3391     0.7431   5.839 5.24e-09 ***
context__WdV    1.9637     0.3313   5.926 3.10e-09 ***
context__Final      NA          NA      NA      NA
```

- Recall that logistic regression fits coefficients a, b, c, d, e to the equation

$$p(\text{reduction}) = \frac{1}{1 + e^{a + b * \text{context} = C + c * \text{context} = __\#C + d * \text{context} = __\#V + e * \text{context} = __\#I}}$$

- Any guesses as to why the computer refused to fit e (“NA”)?
- Work out the predicted probability of $e[s]cuela$.
- Let’s discuss more generally how the regression coefficients relate to the MaxEnt constraint weights.

⇒ In the simple case of a binary outcome, it’s easy to see that MaxEnt and logistic regression are equivalent.

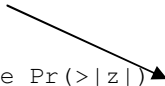
3 Objective functions, loss function

- As Bruce discussed last class, in MaxEnt the weights are fitted so as to maximize the value of a particular *objective function*:
 - maximize predicted probability of observed data (minus a smoothing term—to be discussed later)

- How about in logistic regression?
 - Often in the stats world, people state a function to be minimized rather than one to be maximized
 - The function to be minimized should be some measure of error
 - Do you remember how error is (usually) quantified in linear regression?
 - These functions are typically called *loss functions*
 - This gets at the idea that we might not be interested just in how far off our predictions are, but also in how much those errors “cost”
 - E.g., in medicine both false-negative and false-positive errors are bad, but false-negative is probably worse (serious illness goes untreated, vs. patient is stressed and undergoes further tests).
 - Usual loss function for logistic regression: minimize the negative log probability of the data
 - In other words, same as MaxEnt
 - (as with MaxEnt, though, we’re still ignoring one term in the objective function)

4 Significance

- Advantage of thinking of your problem in logistic regression terms rather than MaxEnt:
 - Your stats software will provide you with a *p* value



	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.4473	0.2050	-2.182	0.0291 *
context__C	3.9234	0.6210	6.318	2.66e-10 ***
context__WdC	4.3391	0.7431	5.839	5.24e-09 ***
context__WdV	1.9637	0.3313	5.926	3.10e-09 ***
context__Final	NA	NA	NA	NA

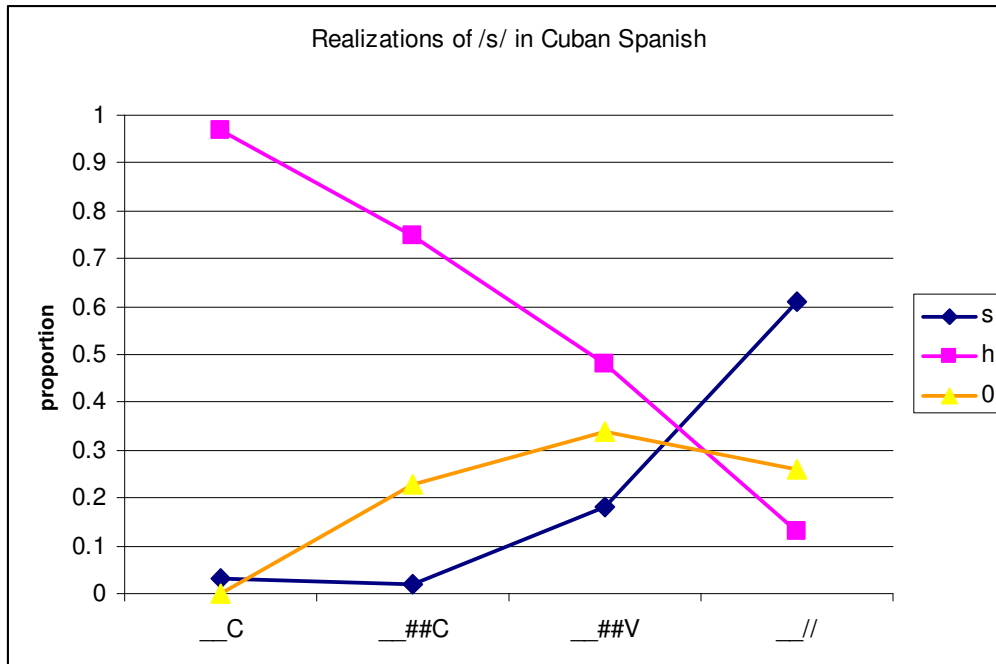
- We’ve seen that “Estimate” is the coefficients in the regression model.
 - “Standard Error”: a property of the coefficients’ covariance matrix (bleh)
 - “z value”: a function of the last two—can you guess what?
 - *p*-value: I believe this is the result of applying a Wald test to the *z* value.
 - “n a set of data where the variable *context__C* actually had no effect, if we drew 100,000 samples and fitted a regression model, how often would we expect *context__C* to get a *z*-value whose absolute value is > 6.318?
- Another method: compare models with and without some factor—but let’s leave that till a later day.

5 By the way...

- Early on, sociolinguistics researchers settled on using logistic regression, sometimes called Varbrul (variable rule) analysis.
- Various researchers, especially David Sankoff, developed software called GoldVarb (Sankoff, Tagliamonte, & Smith 2012 for most recent version) for doing logistic regression in sociolinguistics.
 - Goldvarb uses slightly different terminology though.
 - If you’re reading sociolinguistics work in the Varbrul, see Johnson 2009 for a helpful explanation of how the terminology differs.

6 Multinomial logistic regression

- What if there are 3 possible outcomes, like so:



- We need *multinomial logistic regression*
 - I used `multinom()` here, in the `nnet` package (Venables & Ripley 2002)

Call:

```
multinom(formula = outcome ~ context__C + context__WdC + context__WdV +
  context__Final, data = cuban_ternary)
```

Coefficients:

	(Intercept)	context__C	context__WdC	context__WdV	context__Final
h	1.307279	2.168975	2.317944	-0.3263653	-2.8532748
zero	-1.241953	-7.194329	3.685176	1.8780163	0.3891828

- How to unpack this:
 - First line compares *h* to *faithful* (*faithful* is baseline by default, because first alphabetically)

$$\ln\left(\frac{\text{prob}(\text{outcome} == h)}{\text{prob}(\text{outcome} == \text{faithful})}\right) =$$

$$1.31 + 2.17 * \text{context} == _C + 2.32 * \text{context} == _ \#C - 0.33 * \text{context} == _ \#V - 2.85 * \text{context} == _ //$$

- What is $\ln\left(\frac{\text{prob}(\text{outcome} == h)}{\text{prob}(\text{outcome} == \text{faithful})}\right)$ for the context $_C$?
- Write out what the last line of the R output means.
- Now we have to do some algebra on the board to find $\text{prob}(\text{outcome} == \text{faithful})$
 - Solve the above for $p(\text{outcome} == h)$
 - Solve the above for $p(\text{outcome} == \text{zero})$
 - If the probabilities of the 3 choices must sum to 1, what is $p(\text{outcome} == \text{faithful})$?

- What we should get is: $prob(faithful) = \frac{1}{1 + e^{linear_expression_for_h} + e^{linear_expression_for_zero}}$
- Call the denominator in the expression above Z.
- Then $prob(h) = \frac{1}{Z} e^{line_for_h}$, $prob(zero) = \frac{1}{Z} e^{line_for_zero}$, etc.
 - Determine predicted probability for *h* in __##V

Again, this looks suspiciously like MaxEnt...

7 Comparison to a MaxEnt model

- Training data

			*s	Ident(place)	Max	*hC	*h##C	*h##V	*h//	*0C	*0##C	*0##V	*0//
escuela	e[s]cuela	0.03	1										
	e[h]cuela	0.97		1		1							
	e[]cuela	0			1					1			
dos veces	do[s]	0.02	1										
	do[h]	0.75		1			1						
	do[]	0.23			1						1		
dos equis	do[s]	0.18	1										
	do[h]	0.48		1				1					
	do[]	0.34			1							1	
dos	do[s]	0.61	1										
	do[h]	0.13		1					1				
	do[]	0.26			1								1

- Weights learned

3.624341	*s
0.000000	Ident (place)
1.181994	Max
0.148242	*hC
0.000000	*h##C
2.643512	*h##V
5.170265	*h//
50.000000	*0C
0.000000	*0##C
1.806358	*0##V
3.295124	*0//

- Determine predicted probability for [h] in __##V

⇒ If you look up *Maximum Entropy classifier* in Wikipedia, you're redirected to *Multinomial logistic regression*

- Differences?
 - Translation can be more or less difficult depending on the constraint set—what if instead of markedness constraints for *h* and 0 in each context, there were only a set of markedness constraints for *s* in each context.
 - What if it's not possible to classify outcomes neatly into categories? It's weird to do a multinomial regression with 100 different outcomes, each occurring only once.

8 Fitting and overfitting

- In the MaxEnt model just above, one weight really sticks out—discuss.
- Except for a weight limit of 50, the MaxEnt model above was free to explore the entire space of weights and find the very best fit to the training data—but is this a good thing?
- Suppose we added in some constraints like *h_IFSENTENCE>6WORDS and *s_INSUBORDINATECLAUSE (and train on real corpus data)—discuss what might happen
- In machine learning applications, people worry about **overfitting**. I'll draw some pictures on the board.
 - To summarize what just happened on the board: a model that fits the *existing* data too well could make worse predictions about *new* data.
- One response to overfitting is to do some model comparison to decide if some independent variables should be removed altogether—we'll talk about that another day.
- But another response is to (decide how much to) **penalize** weights/coefficients that are large.
 - We want to trade weight/coefficient size off against fit: in order to have a large coefficient, a constraint/variable should do a lot of work in explaining the data.

9 Smoothing in linear regression

- In simple linear regression, you ask the computer to minimize this error:

$$\sum_{i=1}^n (\text{predicted_value_for_}x_i - \text{actual_value_}y_i)^2$$

- 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:

$$\sum_{i=1}^n (\text{predicted_value_for_}x_i - \text{actual_value_}y_i)^2 + \lambda \sum_{j=1}^m (\text{coefficient}_m)^2$$
 - 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 ?

10 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 . Call it σ_j : $\sum_{i=1}^N \ln P(x_i) - \sum_j \frac{(w_j - \mu_j)^2}{2\sigma_j^2}$
- What would this prior say about these two sets of weights: $\{1,1,1,99\}$, $\{25,25,25,25\}$?
- ⇒ This choice of smoothing term prefers to spread responsibility (weight) evenly across constraints as much as possible.
 - If there are two constraints that could both explain the data, weight them equally rather than just picking one.
- Can you dream up a smoothing term that would have the opposite preference—prefer to pick just one constraint and load all the weight onto it?

11 This smoothing term is often called a Gaussian prior (and it’s not the only choice!)

Why “Gaussian”?

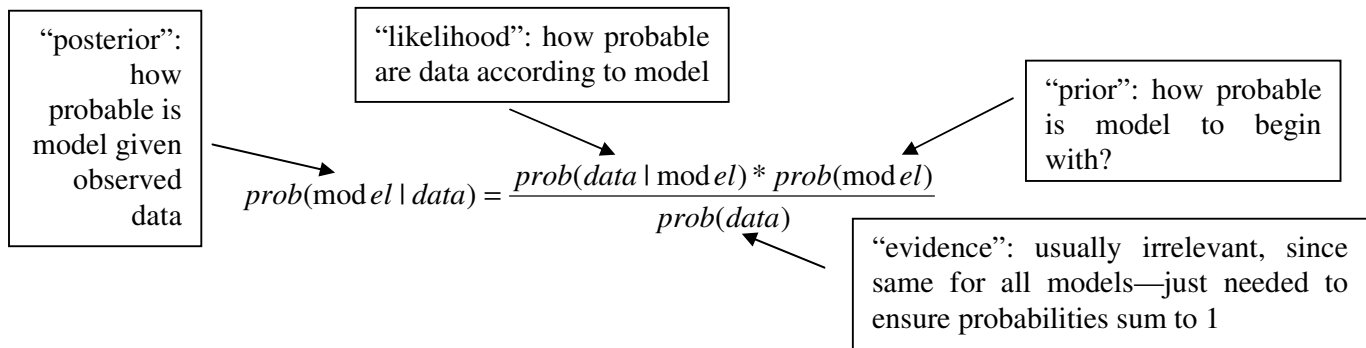
- The equation for the normal distribution, also known as Gaussian distribution, is

$$y = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\text{I'll illustrate this on the board})$$

- Suppose we wanted to maximize: $\ln(\text{prob}(\text{data})) + \sum_{j=1}^M \ln \left(\frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(w_j - \mu_j)^2}{2\sigma_j^2}} \right)$
- i.e., maximize: $\ln(\text{prob}(\text{data})) + \sum_{j=1}^M \left(\ln \left(\frac{1}{\sqrt{2\pi\sigma_j^2}} \right) + \ln \left(e^{-\frac{(w_j - \mu_j)^2}{2\sigma_j^2}} \right) \right) =$
- $\ln(\text{prob}(\text{data})) + \text{a_number_that_doesn't_depend_on_weights} + \sum_{j=1}^M \frac{-(w_j - \mu_j)^2}{2\sigma_j^2}$
- only thing learner can change is weights, so same as maximizing $\ln(\text{prob}(\text{data})) - \sum_{j=1}^M \frac{(w_j - \mu_j)^2}{2\sigma_j^2}$

Why “prior”?

- Recall Bayes’ Law from yesterday’s seminar:



- Taking the log, $\ln p(\text{model}|\text{data}) = \ln p(\text{data}|\text{model}) + \ln p(\text{model}) - \ln p(\text{data})$
 - Compare and contrast this to our MaxEnt objective function with smoothing.

12 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?

References

- Bybee, Joan. 2001. *Phonology and language use*. Cambridge: Cambridge University Press.
- Hooper, Joan Bybee. 1981. The Empirical Determination of Phonological Representations.. In John Laver and John Anderson Terry Myers (ed.), *Advances in Psychology*, vol. Volume 7, 347–357. North-Holland. <http://www.sciencedirect.com/science/article/pii/S0166411508602101> (16 April, 2013).
- Johnson, Daniel Ezra. 2009. Getting off the GoldVarb Standard: Introducing Rbrul for Mixed-Effects Variable Rule Analysis. *Language and Linguistics Compass* 3(1). 359–383. doi:10.1111/j.1749-818X.2008.00108.x (16 April, 2013).
- R Development Core Team. 2010. *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. www.R-project.org.
- Sankoff, David, Sal Tagliamonte & E Smith. 2012. GoldVarb Lion: a multivariate analysis application.. University of Toronto, University of Ottawa, ms. <http://individual.utoronto.ca/tagliamonte/goldvarb.htm>.
- Terrell, Tracy D. 1977. CONSTRAINTS ON THE ASPIRATION AND DELETION OF FINAL /s/ IN CUBAN AND PUERTO RICAN SPANISH. *Bilingual Review / La Revista Bilingüe* 4(1/2). 35–51. doi:10.2307/25743708 (16 April, 2013).
- Terrell, Tracy D. 1979. Final /s/ in Cuban Spanish. *Hispania* 62(4). 599–612. doi:10.2307/340142 (16 April, 2013).
- Venables, W. N. & B. D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. Springer.