Linguistics 251 Spring 2013
Variation in Phonology Hayes/Zuraw

Class 16, 5/23/13: Noisy Harmonic Grammar

1. Assignments etc.

- Hand in your Seuss statistics exercise if you have not already.
- Talk: with us re. projects.

2. How does Noisy Harmonic Grammar work?

- Every constraint has a weight.
- To find a winner, compute dot product of weights and violations for each candidate — roughly, the harmony.
    - Winner = the candidate with the lowest penalty
- To get variation, perturb each weight by a small amount, the **noise**, at **evaluation time**.
- To calculate the probability distribution of outcomes, sample many times over different evaluation times.

3. The proportional analogy from before

Noisy harmonic grammar : nonstochastic harmonic grammar
Stochastic OT : classical OT

4. Who invented it?

- Harmonic grammar itself is from Smolensky and colleagues and dates from the late 1980's if not before.
- Boersma (1998 diss.) thought of "stochasticizing" classical OT with ranking values and noise.
- Noisy harmonic grammar is the obvious intersection of these two ideas.
- I believe the ur-reference is Boersma and Pater (2008, ms. on ROA), which is now revised and destined for publication:
    - Paul Boersma & Joe Pater: Convergence properties of a gradual learning algorithm for Harmonic Grammar. To appear in a book edited by John McCarthy and Joe Pater. On Boersma's web site at www.fon.hum.uva.nl/paul/chrono.html.[1]

[1] I spotted this late in the game so this handout is based on the older version …

HOW CAN I RUN A SIMULATION IN NHG?

5. In OTSoft

- This currently exists in sub-beta form on Bruce's computer; I did the simulations for this class with this version but feel it is not yet publication-quality reliable.
- Ask me for a copy if you like.

6. In Praat

- This is tricky and to my knowledge there is no documentation available.
- I have a set of notes to self on how to do it; it borrows from similar notes by Joe Pater.
- To make the input file: I strongly suggest you do it as an OTSoft file and use the OTSoft conversion utility: **File, Save as Praat**.
    - ➢ It will dump the two files you need for Praat into the output folder for the simulation at hand.
- If you do try to make a Praat file on your own, be warned that in my experience the slightest deviation from required Praat format will cause an undiagnosable crash.

HOW ARE THE CORRECT WEIGHTS LEARNED IN NHG?

7. Source

- Boersma and Pater (2008; 2013)

8. Background

- The method is very similar to the classical Gradual Learning Algorithm for Stochastic OT (but seems to work rather better).

9. "Winners" and "Losers"

- The learning system attends to the ambient data.
- Whatever it hears is paired (somehow) with its input and called a "Winner"
- N.B. if there is free variation, a current "Loser" may actually be a Winner on other occasions; possibly indeed most other occasions. No matter.
- The stochastic grammar is run once on the relevant input and the result checked against the current Winner.
- If the grammar prefers the winner, do nothing.
- If the grammar prefers the loser then see next item.

10. Actions taken when the grammar prefers a loser

- Go through all the constraints.
- If a constraint is **neutral** between winners and losers, do nothing.
- If a constraint **prefers a winner**:

  - Find out *how much* it prefers the winner: subtract winner violations from loser violations.
  - Multiply the result by the **plasticity** (below) and increment the weight of the constraint by this number.
- If a constraint **prefers a loser**:
  - Find out *how much* it prefers the loser: subtract loser violations from winner violations.
  - Multiply the result by the **plasticity** (below) and decrement the weight of the constraint by this number.

11. Putting this all in one formula

$$w_k' = w_k + \varepsilon \cdot \left(s_{pqk} - s_{prk} \right)$$

where

w_k' = new value of weight
w_k = old value of weight
ε = plasticity
s_{pqk} = violations for input p, observed output q, constraint k
s_{prk} = violations for input p, derived output r, constraint k

- This automatically
  - does nothing when a constraint is a non-preferrer
  - adjusts in the correct direction if the constraint is a preferrer.

12. Convergence

- Boersma and Pater provide a convergence proof (the main point of their paper).
- It holds only for non-stochastic training data.
- Results for stochastic continue to be unknown (cf. maxent, where all is known to be fine).

WHAT ARE THE SALIENT BEHAVIORS OF A NOISY HARMONIC GRAMMAR?

13. Ganging is possible

- Since you add up the violations *x* weights, it's possible and normal for two weaker constraints to collectively outweigh a stronger constraint.
- Recall example of Class 4: in Japanese loanword adaptation LYMAN'S LAW and *VOICED GEMINATE gang to outweigh *IDENT(voice).
- Other frameworks that exhibit ganging:
  - Nonstochastic harmonic grammar
  - Maxent
  - (weakly and inconsistently:) stochastic OT

14. Harmonic bounding is enforced

- Suppose Candidate B has a proper superset of Candidate A's violations.
- Then, when we multiply violations *x* weights, we will get a larger result for B — it's the same weight being multiplied for both A and B.
- Therefore, B will be less harmonic than A and will lose.
- N.B. it doesn't matter what the weights are

15. Exceptions to harmonic bounding

- All constraints violated by A and B have weight zero — then we get a tie.
- Negative weights are allowed. On this point, Boersma and Pater cite (Keller 2000, 2006, Prince 2002, and Pater 2009). xxx

16. How negative weights subvert harmonic bounding

- Boerma and Pater's example, reformatted:[2]

Input	Constraint1	Constraint2	Harmony
weight:	1	−2	
Candidate1	1		1
Candidate2		1	−2
Candidate3	1	2	−3

- In Praat, there is a quirky implementation of the "positive only" restriction:
  - When you compute the grammar's output, you take *e* to the constraint weight and use that instead of the constraint weight.
  - This works, since *e* to anything is positive.
  - Obviously, the weights will come out on a different scale.
  - They call this "Exponential Noisy Harmonic Grammar"

17. Negative weights are controversial

- … because of the danger of the infinitely perfect candidate /kæt/ → [kættatatatatatatata…]
- Yet I'm tempted to have them anyway — cf. Seuss, who actively seeks multiple fulfillments of PLEASE HAVE INITIAL [z]. (*Zinzibar-Zanzibar*, *zizzer-zazzer-zuzz*, *zinniga-zanniga*)

18. Multiple-site application: allows/requires independent application

- Let's try a wild guess about how to analyze Makonde, taught last time in Kie's class.

[2] Note: computations a tiny bit different; Boersma and Pater like to use negative numbers for violations, so highest Harmony rather than lowest wins.

- Suppose:
  - assimilation is autosegmental (multiple linking)
  - there is metrical structure in Makonde, assigning prominence to the penult.
- Assume for concreteness that all variants are equiprobable.

			*ATONIC [e]	IDENT(low)	*e unlinked to tonic
kú-pélévéléléela	e e e e ee la	0.20	****		
	a e e e ee la	0.20	***	*	
	a a e e ee la	0.20	**	**	
	a a a e ee la	0.20	*	***	
	a a a a ee la	0.20		****	
	*e a a a ee la		*	***	*

19. Maxent is cool as a cucumber

Constraint	*Weight*
*ATONIC [e]	0
IDENT(low)	0
e UNLINKED TO TONIC	7.23

Input	*Candidate*	*Observed Proportion*	*Predicted proportion*
kú-pélévéléléela	e e e e ee la	0.2	0.200
	a e e e ee la	0.2	0.200
	a a e e ee la	0.2	0.200
	a a a e ee la	0.2	0.200
	a a a a ee la	0.2	0.200
	e a a a ee la	0	0

- Equiprobability follows merely from
  - promoting *E UNLINKED TO TONIC to cover *e a a a ee la
  - letting other constraints null out — the very essence of maximum entropy.

20. Noisy Harmonic Grammar does poorly

```
1. Weights Found

11.023       *Atonic [e]            Note:  these look essentially identical;
10.977       Ident(low)                    input file is symmetrical.
8.000        e unlinked to tonic

2. Matchup to Input Frequencies

   /kú-pélévéléléela/ Input Fr. Gen Fr.  Input #     Gen. #
   e e e e ee la      0.200     0.492    199513       4918
```

```
a e e e ee la      0.200     0.000     200006
a a e e ee la      0.200     0.000     200100
a a a e ee la      0.200     0.000     199946
a a a a ee la      0.200     0.508     200435      5082
e a a a ee la      0.000     0.000
```

/kú-pélévéléléélа/:	*Atonic [e] (11.0227497722459)	Ident(low) (10.9772502277541)	e unlinked to tonic
☞ a a a a ee la (0.508)		****	
a a a e ee la	*	***	
e a a a ee la	*	***	*
a a e e ee la	* *	**	
a e e e ee la	* **	*	
e e e e ee la (0.492)	* ***		

- Reason for failure: at evaluation time, one of the two crucial weights around 11 will be higher; and the candidate that violates it not at all will win.

21. I think this problem can be fixed

- The problem is that noise is attached to *constraints*; but we ought to attach it to *asterisks* or to *tableau cells*.
- See Robert's work for what I believe is a system of this kind:
  - ➢ Goldrick, M. & Daland, R. (2009). Linking speech errors and phonological grammars: Insights from Harmonic Grammar networks. *Phonology* 26: 147-185.
- See also below, on funny sigmoids.

22. Other cases

- I think the English tapping case (*marketability*) would be similar.
- General scheme: *pyramid* against *inverted pyramid* of violations.

HOW DOES NOISY HARMONIC GRAMMAR DO WITH OUR ANTI-GLA EXAMPLES?

23. Recall Class 4

- We attacked the GLA as a system unable to learn the grammars expressible in its theoretical framework.
- We did two examples, one old and one new.

24. The Pater (2008) counterexample

- Scheme: a gnarly set of interlocking patterns, where the same constraint needs to be both promoted and demoted.

			Con1	Con2	Con3	Con4	Con5	Con6
Form1	Winner1	1		1				
	Loser1		1		1			

Form2	Winner2	1			1			
	Loser2			1		1		
Form3	Winner3	1				1		
	Loser3				1		1	
Form4	Winner4	1					1	
	Loser4					1		1
Form5	Winner5	1						1
	Loser5						1	

- GLA fails miserably.
- Maxent does fine (tableaux not repeated here).
- What about NHG? All is well:

```
1. Weights Found

32.469       Con4
25.531       Con5
24.000       Con3
10.469       Con6
10.000       Con2
8.000        Con1

2. Matchup to Input Frequencies

   /Form1/   Input Fr. Gen Fr.  Input #     Gen. #
   Winner1     1.000   1.000    2001630      10000
   Loser1      0.000   0.000

   /Form2/   Input Fr. Gen Fr.  Input #     Gen. #
   Winner2     1.000   1.000    1998980      10000
   Loser2      0.000   0.000

   /Form3/   Input Fr. Gen Fr.  Input #     Gen. #
   Winner3     1.000   1.000    1999833      10000
   Loser3      0.000   0.000

   /Form4/   Input Fr. Gen Fr.  Input #     Gen. #
   Winner4     1.000   1.000    1999763      10000
   Loser4      0.000   0.000

   /Form5/   Input Fr. Gen Fr.  Input #     Gen. #
   Winner5     1.000   1.000    1999794      10000
   Loser5      0.000   0.000
```

25. The descent-into-hell example

- Scenario: a modest degree of lexical "incongruity"
- Constraint C is accidentally a tiny-bit dysfunctional in one common location; crucial in another rarer location.
- Input file:

			Prefer A	Prefer B	Prefer C	Prefer D

AB forward	A	510		*	*	
	B	490	*			
AB backward	A	490		*		
	B	510	*			
Bystander	C	2				*
	D	8			*	

- GLA does horribly.
  - C, dysfunctional in common inputs, is sent down to hell by endless stochastic demotions.
  - The descent happens so fast that D can't keep up.

```
1. Ranking Values Found

PreferA     100.053
PreferB     99.947
PreferD     -503.031
PreferC     -2,686.638

2. Matchup to Input Frequencies


   /Frequent-C/ Input Fr. Gen Fr.  Input #     Gen. #
   A with C       0.510   0.514     254046      51433
   B with C       0.490   0.486     244063      48567

   /Frequent-no C/ Input Fr. Gen Fr.  Input #     Gen. #
   B no C         0.510   0.486     253380      48567
   A no C         0.490   0.514     243465      51433

   /Rare input/ Input Fr. Gen Fr.  Input #     Gen. #
   D              0.800   1.000       4011     100000
   C              0.200   0.000       1035
```

- Maxent does fine:

```
1. Constraints and weights

   0.984  PreferA
   0.984  PreferB
   0.000  PreferC
   1.386  PreferD

2. Inputs, candidates, input frequencies, input proportions, predicted
probabilities

Inputs      Candidates  Input frequencies  Input proportions  Predicted
Frequent-C   A with C          510                  0.510        0.500
             B with C          490                  0.490        0.500


Frequent-no C    A no C           490                  0.490     0.500
                 B no C           510                  0.510     0.500


Rare input      C               2                   0.200        0.200
                D               8                   0.800        0.800
```

- Observe: best-available compromise upstairs, perfect fit downstairs.
- How did it succeed?
- C is worthless among frequent forms, and gets a zero to maximize fit.
- What about the rare forms? No problem — maxent doesn't need two conflicting constraints to get variation! PREFER D does all the work.
- … perhaps you see the problem that is lurking on the horizon …

- NHG: it seems a bit unstable and inaccurate, but with enough learning trials and a tiny final plasticity things are more or less ok:

```
1. Weights Found

6.104        PreferA
5.896        PreferB
2.446        PreferD
0.030        PreferC

2. Matchup to Input Frequencies

   /Frequent-C/ Input Fr. Gen Fr.  Input #     Gen. #
   A with C       0.510   0.522    1269715      52247
   B with C       0.490   0.478    1220147      47753

   /Frequent-no C/ Input Fr. Gen Fr.  Input #     Gen. #
   B no C         0.510   0.468    1267487      46820
   A no C         0.490   0.532    1217633      53180

   /Rare input/ Input Fr. Gen Fr.  Input #     Gen. #
   D              0.800   0.803      19915      80274
   C              0.200   0.197       5103      19726
```

- Observe that just as in maxent, the weight of PREFER C is essentially zero.

26. Excursus: the worm in the apple

- I believe that the success of maxent and NHG (the two harmonic grammars) is essentially **accidental**.
- Let's **flip the frequencies** for the rare input and see what happens.

			PREFER A	PREFER B	PREFER C	PREFER D
AB forward	A	510		*	*	
	B	490	*			
AB backward	A	490		*		
	B	510	*			
Bystander	C	**8**				*
	D	**2**			*	

- GLA: still a problem; D can't catch up with C (indeed, this time it must go *lower* than C to get the right proportions), so error occurs:

```
1. Ranking Values Found

PreferB     100.125
PreferA     99.875
PreferD     -1,037.730
PreferC     -1,040.098

2. Matchup to Input Frequencies

   /Frequent-C/ Input Fr. Gen Fr.  Input #     Gen. #
   A with C       0.510   0.464     253732      46362
   B with C       0.490   0.536     243830      53638

   /Frequent-no C/ Input Fr. Gen Fr.  Input #     Gen. #
   B no C         0.510   0.536     253568      53638
   A no C         0.490   0.464     243820      46362

   /Rare input/ Input Fr. Gen Fr.  Input #     Gen. #
   C              0.800   0.200       4040      19952
   D              0.200   0.800       1010      80048
```

- Maxent flatlines! Hence a heat-death-of-universe output pattern.

Constraint	Mu	Sigma	Weight
PreferA	0	100000	**0**
PreferB	0	100000	**0**
PreferC	0	100000	**0**
PreferD	0	100000	**0**

Input	Candidate	Observed	Obs. Prop	Predicted
Frequent-C	A with C	510	0.51	0.5
	B with C	490	0.49	0.5
Frequent-no C	A no C	490	0.49	0.5
	B no C	510	0.51	0.5
Rare input	C	8	0.8	0.5
	D	2	0.2	0.5

- What's going on?
- Null weights for the frequent input are rational — they match 50/50 (as good as it can be matched, given the constraint set) and obey the Gaussian prior.
- Nulls for the rare input: you'd need a positive weight for Prefer C — but *due to constraint ganging* any positive weight for Prefer C would badly mess up the results on the frequent input.
- Maxent's best bet: **punt** the hope of fitting the infrequent input (50/50 is the best you can achieve) in order to preserve 50/50 on the frequent input.
- Upshot: *this is a problem with the theory, not the learning algorithm.*[3]
  - — No correct analysis available.
  - — If such example exist in the real world, maxent is wrong.
- Bear in mind: there *is* a stochastic OT grammar that generates the data!

---

[3] It could not be a problem with the learning algorithm, which has been proven to be perfect.

— So GLA has a learning problem, maxent a theory problem — if real cases exist.

27. Excursus II: how does NHG do on this worm?

- Again, learning seems a bit unstable and "jiggly"; different runs yield different results.
- Here is a run in which Prefer C and Prefer D flatlined; others were close.

```
1. Weights Found

6.092        PreferA
5.908        PreferB
0.000        PreferC
0.000        PreferD

2. Matchup to Input Frequencies

   /Frequent-C/ Input Fr. Gen Fr.  Input #     Gen. #
   A with C       0.510   0.521    2538056      52079
   B with C       0.490   0.479    2440870      47921

   /Frequent-no C/ Input Fr. Gen Fr.  Input #     Gen. #
   B no C         0.510   0.475    2534992      47515
   A no C         0.490   0.525    2436112      52485

   /Rare input/ Input Fr. Gen Fr.  Input #     Gen. #
   C              0.800   0.501      39764      50089
   D              0.200   0.499      10206      49911
```

28. Upshot

- What I thought was another argument against the GLA is in fact a mixed bag:
  - An argument against the GLA
  - A potential problem for both varieties of harmonic grammar: constraints that are noise in big areas of the data cannot then be used to regular little areas of the data.
- Bear in mind that in actual analytic practice with GLA, descent-into-hell seems to happen quite a lot; so this all may be more than idle speculation.

SIGMOIDS IN NHG

29. Theme

- We've been keeping our eyes open for sigmoids in grammar.
  - Maxent and Stochastic OT naturally derive them when a **variable** constraint conflicts with an **invariant** one.
  - Example we did: Tommo So vowels harmony, with variable AGREE (distance from stem) vs. invariant IDENT.
- What about Noisy Harmonic Grammar?

30. A working example: tone perception

- We want a really really simple example
- Let's do a "perception grammar" in the style of Boersma.
- Two phonemic tones, H and L.
- For this speaker/context/style, the target value for L is 120 Hz.
- Input is an F0 value, output is a probability distribution over H and L.
- Two constraints:
  - WHAT YOU ARE HEARING IS L. Violated by the H hypothesis.
  - *BAD TOKEN OF L. Violated by the L hypothesis: once for every hert that signal is above 120 Hz.

Input: 127 hz.	WHAT YOU ARE HEARING IS L	*BAD TOKEN OF L
L		7
H	1	

- Let us derive a function that relates Hz. above 120 to probability of the perception of L.

31. Warmup: Doing the problem in maxent

- By the maxent formula, if pitch is *n* Hz above 120, then we have:

$$P(L) = e^{-n\,*\text{WBadToken}} / (e^{-n\,*\text{W.BadToken}} + e^{-\text{W.HearL}})$$

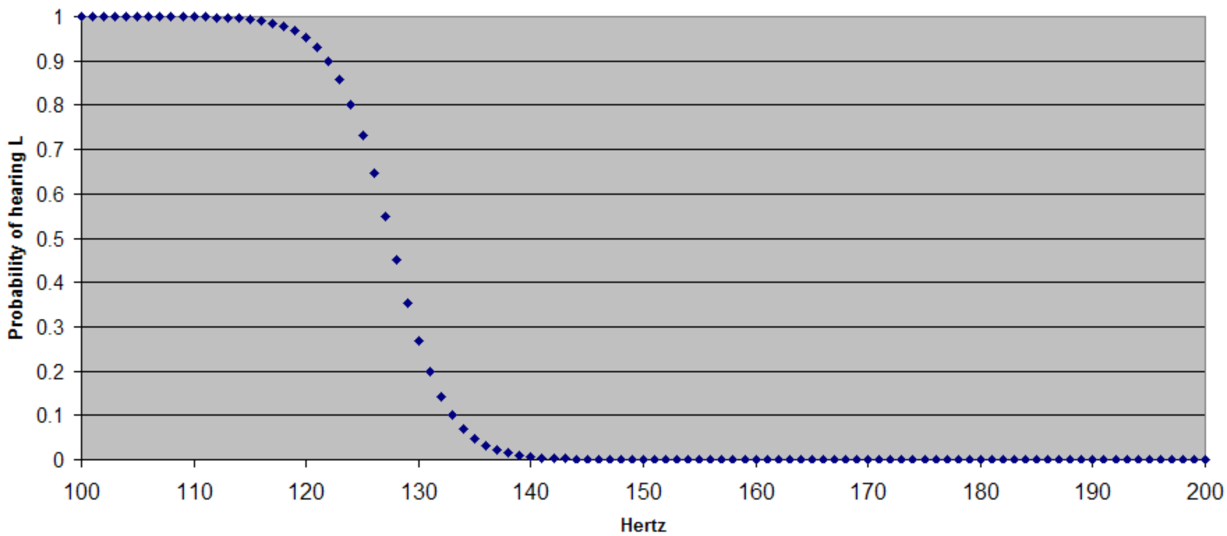
- By algebra we've done before, this is transformed to:

$$P(L) = 1 / 1 + (e^{(n*\text{W.BadToken} - \text{W.HearL}})$$

$$= 1 / 1 + (e^{\text{W.BadToken}\,*(n - \text{W.HearL/ W.BadToken}})$$

- This is a sigmoid that
  - crosses the 50/50 probability line at n = W.HEARL/W.BADTOKEN
  - has steepest slope at that point

32. Implementing this in Excel



- This is the curve for (arbitrarily chosen)
  - w.BadTokenOfL = .4
  - w.HearL = 3
- The crossing point of 50/50 is indeed 120 + 3/.4 = 127.5

33. What about Noisy Harmonic Grammar?

- Consider an even simpler case — two constraints; how does probability vary with the difference in their weights?

Input	Constraint1	Constraint2
weight:	x	y
Candidate1	1	
Candidate2		1

- Since this is just a two-constraint grammar, Candidate1 will win if *at evaluation time*, the perturbed values are such that Constraint2 gets a higher perturbed value than Constraint1.
- By hypothesis, the perturbed values are drawn from normal distributions P1 and P2 with means at x and y.
  - For convenience, let's assume these normal distributions have a variance of 1.[4]
- Now, consider a new probability distribution PD — the probability distribution for the *difference* of P1 and P2.
- Reference sources[5] will tell us that the probability distribution of the difference between two normal distributions has these properties:

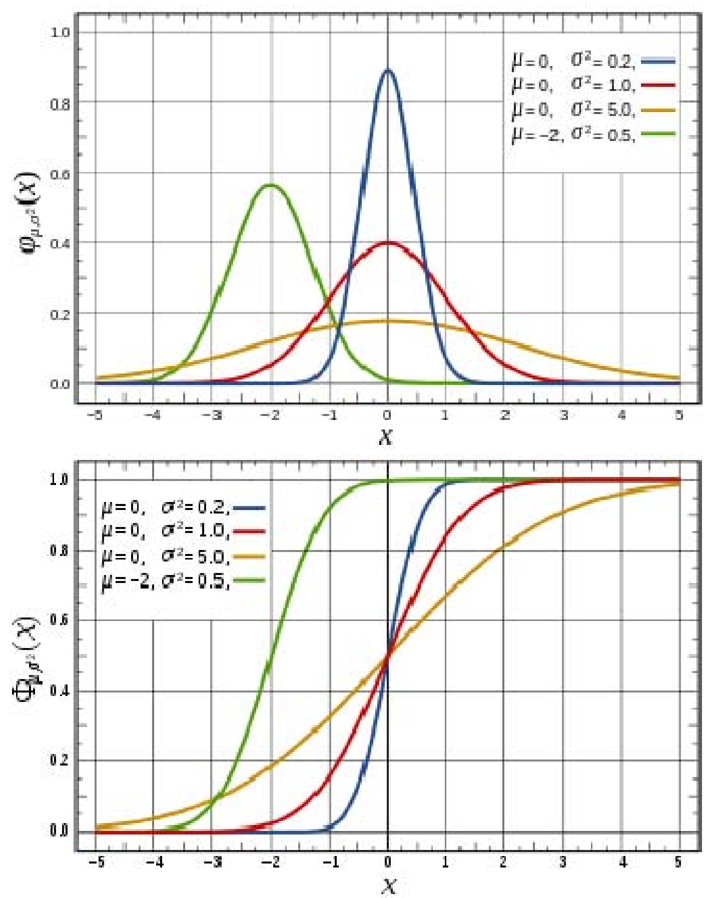
---

[4] Bigger variance means broader curve.
[5] E.g.
http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&ved=0CE0QFjAF&url=http%3A%2F%2Fwww.baradene.school.nz%2FModules%2FResources%2FDownload.aspx%3FID%3

- Mean is difference of the means, so $x - y$.
- Variance is sum of the variances.

- So, at evaluation time, what is the probability that the perturbed value for Constraint2 will be greater than the perturbed value for Constraint1? We look at PD and check the probability that it is greater than zero.
- This is done by integrating under the curve in the zone above zero.
- Appallingly, the normal distribution cannot be integrated, but people have gotten used to this … the result is called the cumulative normal distribution.
- It's intuitive, though, that the cumulative normal distribution would be a sigmoid — slope goes up as you go "over the hump" of the normal distribution.

34. Wikipedia's picture of the normal distribution and its integral, the cumulative normal distribution

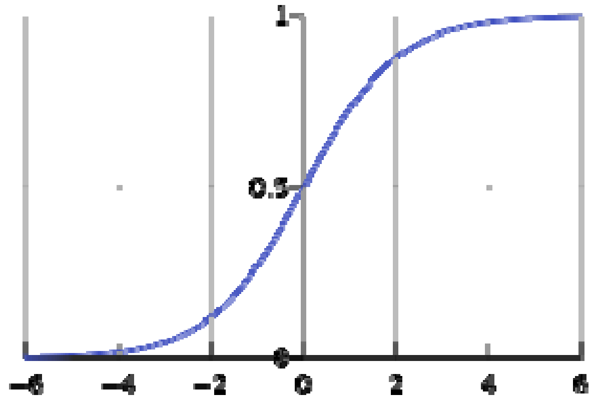


- High variances in the Gaussian go with shallower slopes in the sigmoid.
- μ determines where the sigmoid crosses the 50/50 line.

D6719&ei=SQydUZSqB7T9iQKjy4CIBQ&usg=AFQjCNF7q6RGkAj6UK74p_Zfj4y1HHKnGQ&bvm=bv.46751780,d.cGE

35. Comparison with maxent

- Its sigmoid, covered above, is called the **logistic function** and it looks like this:



- It's pretty clear that linguistic variation data would be unlikely to be precise enough to distinguish between these curves.

36. Upshot so far

- Noisy Harmonic Grammar has at least the propensity to derive sigmoids; seen when we simply vary the distance between the weights of two conflicting constraints.

37. L tone perception in NHG?

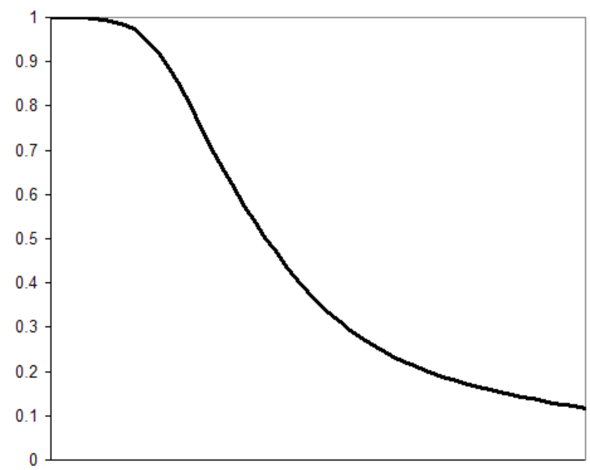
- I ought to give you a Noisy Harmonic Grammar version of the speech perception simulation but I'm not sure how and am out of prep time…

38. The complication: warped sigmoids

- At least in its orthodox version, you
  - Add the noise to the constraint weight first
  - So, when you multiply weights by violations, you are:
  - multiplying the noise
  - So, the two normal distributions being sampled from have different variances!
  - … which means that when one constraint can have multiple violations, the sigmoid is warped.

39. A warped sigmoid for Tommo So

- from McPherson and Hayes (in progress); a purely schematic (but authentically derived) warped sigmoid:



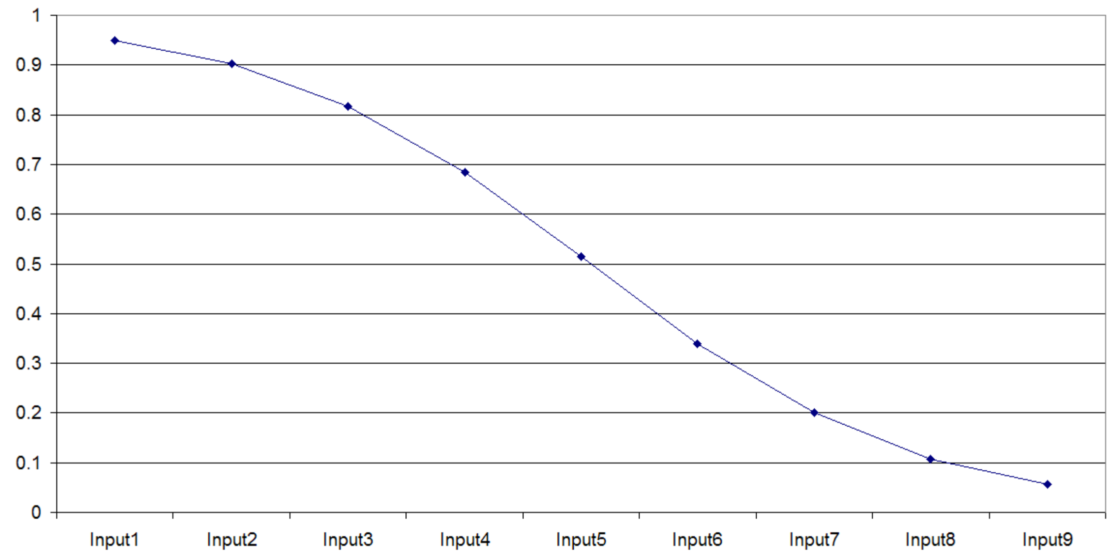
40. A comparison between warped (NHG) and symmetrical (Maxent) distributions

- I asked, if you have a continuum with 95/5 on one end and 5/95 on the other, how would the two theories interpolate?
- Input file, a sort of 9-category Tommo So:

			Faith	Agree by distance
Input1	harmony	95	1	
	no harmony	5		8
Input2	harmony		1	
	no harmony			7
Input3	harmony		1	
	no harmony			6
Input4	harmony		1	
	no harmony			5
Input5	harmony		1	
	no harmony			4
Input6	harmony		1	
	no harmony			3
Input7	harmony		1	
	no harmony			2
Input8	harmony		1	
	no harmony			1
Input9	harmony	5	1	
	no harmony	95		0

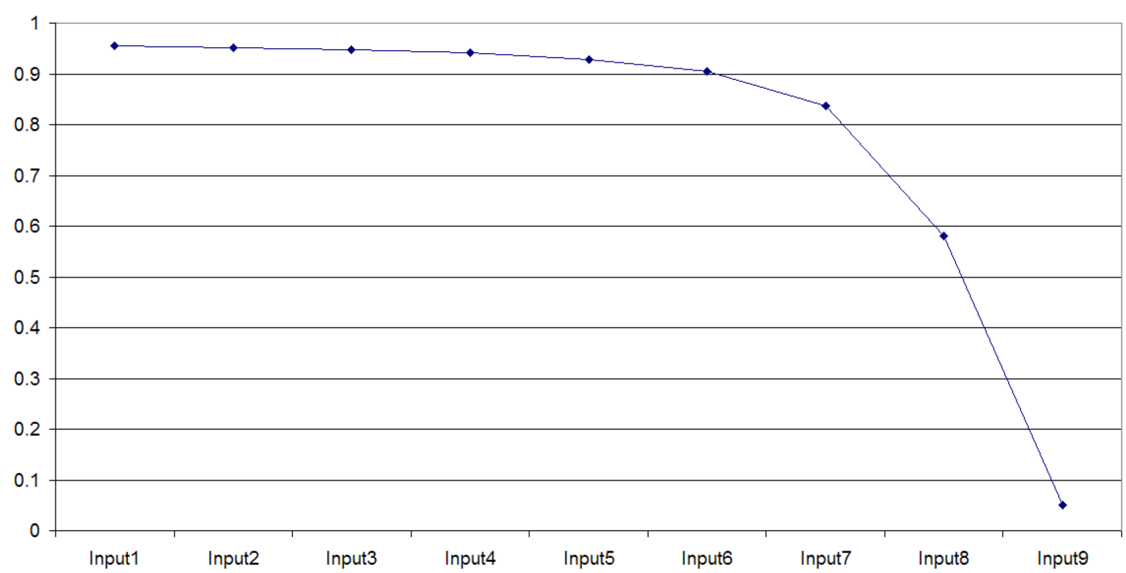
41. A maxent grammar

- learned by Maxent Grammar Tool with $\sigma = 10$:
- Vertical axis is predicted probability
- Observe that the termini were learned accurately
- Weights: Faith 2.82, Agree .72



42. An NHG grammar

- Learned in OTSoft
- Observe that the termini were learned accurately
- Weights: Faith 3.2, Agree 3.8



- This is surely the left side of a heavily warped sigmoid, gentle on left and sharp on right (it's clearly going to level off at zero pretty soon!).

43. Should be we using warped sigmoids? How to tell?

- language data with continua stacked against invariant constraints
- we could do experiments on how people interpolate.

44. The content of NHG is negotiable

- If asymmetrical sigmoids are problematic, we might readjust how noise works: put it on the cells of the tableaux rather than the constraints.
- Then — I think — noise doesn't get multiplied, and the sigmoids will become symmetrical (with the cumulative normal distribution)

EMPIRICAL WORK IN NHG

45. Caveat

- I suspect most studies have not tried out both kinds of harmonic grammar.
- Probably the most important element of a model is whether you use harmonic grammar at all in some form.
- In my experience: maxent and NHG models come out looking very similar (except the case just done, which was fictional…).

46. Some work cited by Boersma and Pater (2013)

- This is probably the biggie:

Jesney, Karen, and Anne-Michele Tessier. 2011. Biases in Harmonic Grammar: The road to restrictive learning. *Natural Language and Linguistic Theory* 29:251--290.

Bottom line: using NHG instead of stochastic OT gives you a beneficial automatical learning bias; i.e. favoring of special-context faithfulness constraints.

- Another item I really like as a poster:

Jesney, Karen. 2007. The locus of variation in weighted constraint grammars. Poster presented at the *Workshop on Variation, Gradience and Frequency in Phonology*. Stanford University, July 2007. [Available at http://www.stanford.edu/dept/linguistics/linginst/nsf-workshop/Jesney_Poster.pdf]

careful and thoughtful framework-comparison

- A tiny, tiny empirical difference:

Hayes, Bruce and Claire Moore-Cantwell (2011) Gerard Manley Hopkins's sprung rhythm: corpus study and stochastic grammar. *Phonology* 28:235–282.

We tried both and got similar results.
But there may be one or two cases in which Hopkins chose to employ a harmonically-bounded winner — possible only in maxent.