Directionality Effects via Distance-based Penalty Scaling

Sharon Inkelas and Eric Wilbanks

UC Berkeley

September 17, 2017

Annual Meeting on Phonology – New York University

Introduction

Directionality

- Proposal: distance based penalty scaling
- Straightforwardly models directionality effects, which have long been a challenge for surface-oriented constraint-based models.
- Our analysis unifies previously unconnected phenomena: directionality, gradient and categorical edge effects, and metrical windows.
- Builds upon recent work in cell multipliers (Boersma & Pater 2016), linear scaling (Hsu & Jesney 2016, McPherson & Hayes 2016, Stanton 2016), and non-linear scaling (Zymet 2016).

- Proposal: distance-based penalty scaling
    - Nkore-Kiga sibilant harmony
- Edge Effects
    - Japanese mimetic palatalization
    - Selkup stress
    - Chaha palatalization/labialization
- Window Effects
    - Metrical window typology

Proposal

Proposal

- We propose distance-based penalty scaling of constraints.

Proposal

- We propose distance-based penalty scaling of constraints.
- Constraint violations are scaled by the distance between the locus of violation and some designated boundary.

Proposal

- We propose distance-based penalty scaling of constraints.
- Constraint violations are scaled by the distance between the locus of violation and some designated boundary.
- This intuition underlies the family of gradient Align constraints (McCarthy & Prince 1994); our innovation is applying this mechanism to Faithfulness and Markedness constraints

Proposal

- We propose distance-based penalty scaling of constraints.
- Constraint violations are scaled by the distance between the locus of violation and some designated boundary.
- This intuition underlies the family of gradient Align constraints (McCarthy & Prince 1994); our innovation is applying this mechanism to Faithfulness and Markedness constraints
- (Open question: to what extent is the Align family still needed?)

Example Illustration

Toy example: sibilants assimilate in [anterior] to the input value of the rightmost sibilant in the word. Edge distance is assessed on the sibilant projection (see e.g. Hansson 2014), with the lowest value being 1:

Example Illustration

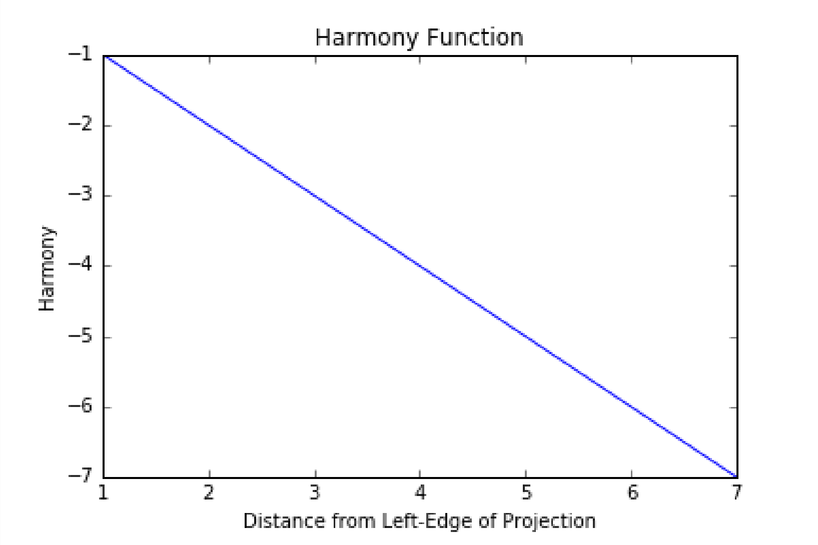
Toy example: sibilants assimilate in [anterior] to the input value of the rightmost sibilant in the word. Edge distance is assessed on the sibilant projection (see e.g. Hansson 2014), with the lowest value being 1:

	/aʃasa/	CORR-SS, ID-SS-ant	ID-IO-ant *(x left-index)*	
		2	*1*	$\mathcal{H}$
a.	aʃasa	-1		-2
☞ b.	asasa		-1 *x 1* = **-1**	-1
c.	aʃaʃa		-1 *x 2* = **-2**	-2

Table 1: Regressive sibilant harmony

To achieve directionality, we scale the penalties for IDENT-IO-ant violations according to distance from the left edge of the sibilant projection. In this simple example, the multiplier for violations in the nth sibilant, counting from the left, is *n*.

Example Harmony Function



Linear Scaling

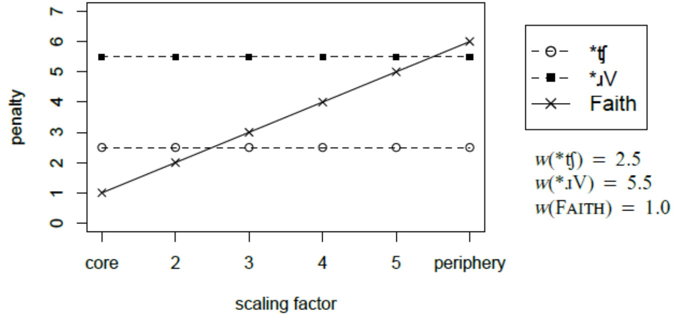


Figure 2: Pattern generated with a single scaled faithfulness constraint.

Hsu and Jesney (2016) take a similar approach to modeling stratum effects in loanword adaptation.

In strata 1-2, both illicit sequences are repaired for native French phonology.

In strata 3-5 only *ɹV is repaired.

In stratum 6 nothing is repaired and the fully faithful loanword is produced.

Nkore-Kiga

Real example: anticipatory sibilant harmony in Nkore-Kiga (Bantu, Uganda).

Nkore-Kiga

Real example: anticipatory sibilant harmony in Nkore-Kiga (Bantu, Uganda).

In Nkore-Kiga (NK), [s] and [ʃ] are in complementary distribution: [s] occurs before [i], and [ʃ] occurs before [e,a,u,o] ("S-MARKEDNESS")

Nkore-Kiga

Real example: anticipatory sibilant harmony in Nkore-Kiga (Bantu, Uganda).

In Nkore-Kiga (NK), [s] and [ʃ] are in complementary distribution: [s] occurs before [i], and [ʃ] occurs before [e,a,u,o] ("S-MARKEDNESS")

This distribution is disrupted when there is more than one sibilant in a word. In that case, all sibilants must agree in [anterior]. Importantly, the harmony process is governed by the surface form of the rightmost sibilant.

Nkore-Kiga

Real example: anticipatory sibilant harmony in Nkore-Kiga (Bantu, Uganda).

In Nkore-Kiga (NK), [s] and [ʃ] are in complementary distribution: [s] occurs before [i], and [ʃ] occurs before [e,a,u,o] ("S-MARKEDNESS")

This distribution is disrupted when there is more than one sibilant in a word. In that case, all sibilants must agree in [anterior]. Importantly, the harmony process is governed by the surface form of the rightmost sibilant.

a.	/-SaːS-ire/	saːsire	(*ʃaːsire)	'be in pain (perf.)'	(cf. *ʃaːʃ-a* 'be in pain')
b.	/-SiːS-a/	ʃiːʃa	(*siːʃa)	'do wrong, sin'	

Nkore-Kiga Scaling

	/-SaːS-ire/	CORR-SS, ID-CC-ant	S-MARKEDNESS *(x left-index)*	
		2	*1*	$\mathcal{H}$
a.	ʃaːsire	-1		-2
☞ b.	saːsire		(-1 *x 1*) = **-1**	-1
c.	ʃaːʃire		(-1 *x 2*) = **-2**	-2

Table 2: Nkore-Kiga sibilant harmony

Nkore-Kiga: challenge for previous accounts

Nkore-Kiga presents a challenge to existing approaches to directionality (Hansson 2001, Bennett & Pulleyblank (to appear)).

- Given two non-agreeing sibilants, such that one must violate a local phonotactic constraint in order for both to agree, string-internal agreement constraints can require that both consonants are the same. But they cannot determine which one should change.
- The decision is not values-based; both [s...s] and [ʃ...ʃ] are possible outcomes.
- The decision is not based on (relative or positional) faithfulness; this is a purely allophonic alternation.
- The decision is not based on absolute position; standard positional markedness (*e.g.*, Zoll 1997) will not suffice. Rather, it is the case that the consonant which obeys local phonotactics, and triggers agreement on the part of the other consonant, is the one which is *closer* to the right edge.

- Solution: distance-based penalty scaling for the local markedness constraints *sa and *ʃi.
- (See also Hansson 2001 for a proposal involving targeted constraints, and Bennett & Pulleyblank (to appear) for a proposal to condition the alternation morphologically.)

Edge Effects

Conflicting Directionality

- Distance-based scaling can also be recruited in the analysis of edge effects, including those involving conflicting directionality ("opposite-edge effects"; see *e.g.* Zoll 1997, Alber 2005)

Conflicting Directionality

- Distance-based scaling can also be recruited in the analysis of edge effects, including those involving conflicting directionality ("opposite-edge effects"; see *e.g.* Zoll 1997, Alber 2005)
- Let's consider one opposite-edge effect: Japanese mimetic palatalization.

Japanese Palatalization

The descriptive facts of Japanese mimetic palatalization (as drawn from Mester & Ito 1989) are as follows:[1]

[1]Setting aside complications of /r/.

Japanese Palatalization

The descriptive facts of Japanese mimetic palatalization (as drawn from Mester & Ito 1989) are as follows:[1]

- palatalize a coronal in a coronal-noncoronal sequence

[1] Setting aside complications of /r/.

Japanese Palatalization

The descriptive facts of Japanese mimetic palatalization (as drawn from Mester & Ito 1989) are as follows:[1]

- palatalize a coronal in a coronal-noncoronal sequence
- palatalize the **right-most coronal**

[1] Setting aside complications of /r/.

Japanese Palatalization

The descriptive facts of Japanese mimetic palatalization (as drawn from Mester & Ito 1989) are as follows:[1]

- palatalize a coronal in a coronal-noncoronal sequence
- palatalize the **right-most coronal**
- if no coronals, the **left-most noncoronal** is palatalized

[1] Setting aside complications of /r/.

Japanese Scaling

	/ptptp/	HAVEPAL	*p^j *(left-index)*	ID *(right-index)*	
		7	*2*	*1*	$\mathcal{H}$
a.	ptptp	-1			-7
b.	ptptp^j		(-1 x *3*) = **-3**	(-1 x *1*) = **-1**	-7
→ c.	ptpt^{j}p			(-1 x *2*) = **-2**	-2
d.	pt^{j}ptp			(-1 x *4*) = **-4**	-4
e.	p^jtptp		(-1 x *1*) = **-1**	(-1 x *5*) = **-5**	-7

Table 3: Japanese Mimetic Palatalization: Coronals

Left-index: worse violations farther from left-edge

Right-index: worse violations farther from right-edge

Japanese Scaling

	/pppp/	HAVEPAL	*p^j *(left-index)*	ID *(right-index)*	
		7	*2*	*1*	$\mathcal{H}$
a.	pppp	-1			-7
b.	$pppp^j$		(-1 *x* *4*) = **-4**	(-1 *x* *1*) = **-1**	-9
→ c.	p^jppp		(-1 *x* *1*) = **-1**	(-1 *x* *4*) = **-4**	-6

Table 4: Japanese Mimetic Palatalization: No Coronals

Selkup Stress

Another case of conflicting directionality is Selkup stress which can be described generally as follows (Zoll 1997):

- Rightmost heavy syllable receives stress
- If no heavy syllables, leftmost light receives stress

Selkup Stress

Another case of conflicting directionality is Selkup stress which can be described generally as follows (Zoll 1997):

- Rightmost heavy syllable receives stress
- If no heavy syllables, leftmost light receives stress

(Data from Halle & Clements 1983)

uːˈcɔːmɨt – ‘we work’ (rightmost heavy)

ˈkarman – ‘pocket’ (leftmost light)

Chaha Palatalization/Labialization

- What we've seen so far: linear scaling functions
- By including non-linear scaling, we can derive absolute edge effects
- Chaha impersonal forms demonstrate both gradient and absolute edge-tropism
- Our solution: two scaling functions, one linear and one non-linear

The basic descriptive facts of Chaha are as follows (Leslau 1967, McCarthy 1983; see Banksira 2000, Rose 2007 for additional nuances)

- To encode perfective, third person masculine singular object, labialize rightmost noncoronal, which can be final (a) or nonfinal (b,c) (Data from McCarthy 1983:179):

a.	‘find’	nækaeb	$\longrightarrow$	nækæb^{w}
b.	‘lack’	bækær	$\longrightarrow$	bæk^{w}ær
c.	’seem’	mæsær	$\longrightarrow$	m^{w}æsær

- This is accomplished by a simple linear penalty scaling function, indexed to the right edge of the word.

Basic descriptive facts of Chaha, cont.

- To encode feminine gender imperatives: palatalize a coronal in absolute final position (a). If stem doesn't end in coronal, interior coronals are not palatalized (b):

a.	'bite'	nəkəs	$\longrightarrow$	nəkəs^{y}	McCarthy 1983:179
b.	'hash'	kɨtɨf	$\longrightarrow$	kɨtɨf	Rose 2007

- Modeling the absolute-edge effect requires a different scaling function: a right-indexed, **nonlinear** function in which the penalty increases so sharply between distance points 1 and 2 that the cost of palatalizing a nonfinal coronal outweighs the benefit of obeying the requirement to palatalize

Basic descriptive facts of Chaha, cont.

- To encode impersonal, both palatalization and labialization processes apply. Thus we observe the two scaling functions co-occurring in the analysis of the same word

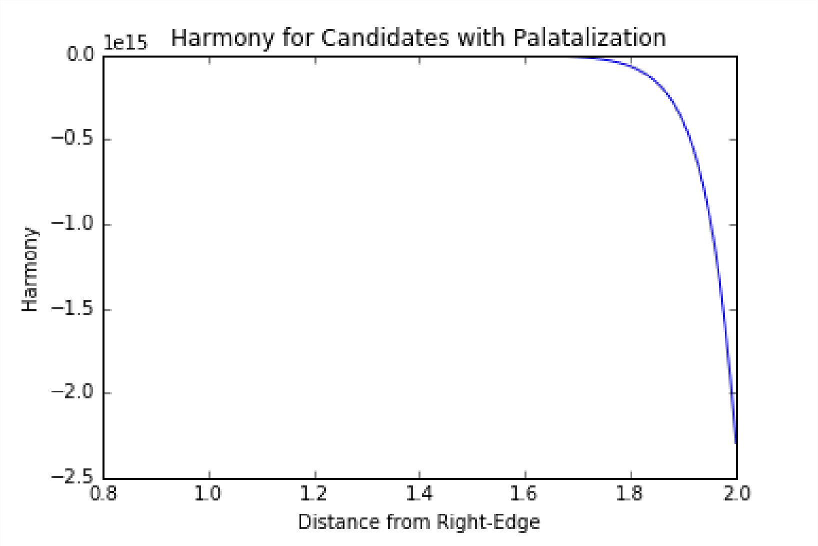
Modeling Chaha Impersonal Forms

“open”	kæfæt	Have Lab.	Have Pal.	$*C^j$ (R-index; exp.)	$*C^w$ (R-index; linear)	
		w	x	y	z	$\mathcal{H}$
01	kæfæt	-1	-1	0	0	
02	kæfæt^j	-1	0	-1*f(1)	0	
03	→ kæf^{w}æt^j	0	0	-1*f(1)	-1*g(3)	
04	kæf^{w}æt	0	-1	0	-1*g(3)	

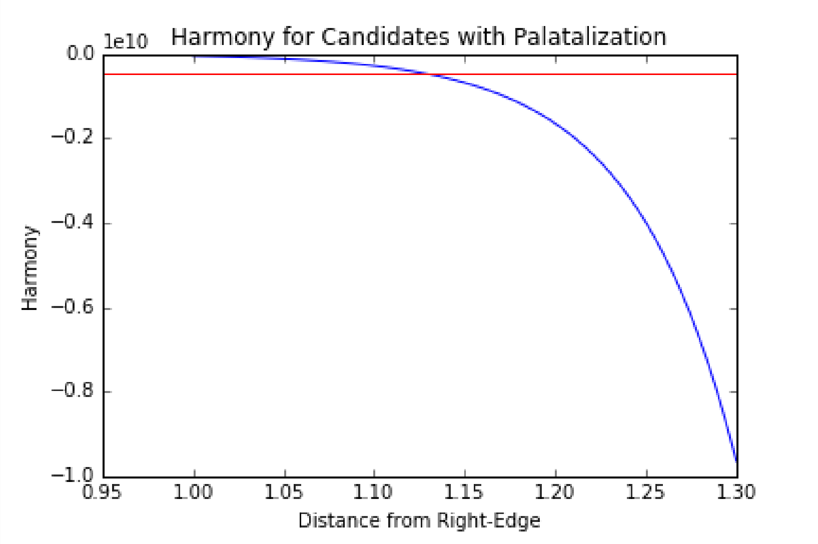
“kill”	qætær	Have Lab.	Have Pal.	$*C^j$ (R-index; exp.)	$*C^w$ (R-index; linear)	
		w	x	y	z	$\mathcal{H}$
05	qætær	-1	-1	0	0	
06	qæt^{j}ær	-1	0	-1*f(3)	0	
07	q^{w}æt^{j}ær	0	0	-1*f(3)	-1*g(5)	
08	→ q^{w}ætær	0	-1	0	-1*g(5)	

Where $f(n) = a^n$ and $g(n) = b * n$.

Absolute edge-effects via non-linear scaling



Absolute edge-effects via non-linear scaling



Windows

Deriving Windows

As we've seen, distance-based scaling can efficiently model a variety of directional effects.

Deriving Windows

As we've seen, distance-based scaling can efficiently model a variety of directional effects.

The nonlinear function used for Chaha derived absolute-edge effects can be extended to the analysis of windows.

Deriving Windows

As we've seen, distance-based scaling can efficiently model a variety of directional effects.

The nonlinear function used for Chaha derived absolute-edge effects can be extended to the analysis of windows.

We extend our proposal to the analysis of initial or final windows, which are captured in our model by the same mechanism as absolute-edge effects. The difference between these two phenomena lies only in the inflection point of the non-linear function.

Deriving Windows

As we've seen, distance-based scaling can efficiently model a variety of directional effects.

The nonlinear function used for Chaha derived absolute-edge effects can be extended to the analysis of windows.

We extend our proposal to the analysis of initial or final windows, which are captured in our model by the same mechanism as absolute-edge effects. The difference between these two phenomena lies only in the inflection point of the non-linear function.

Here we focus on analyses of metrical windows, the most well-known example of the phenomenon.

Metrical Windows

For a language with (metrical) windows, primary stress may only fall within a certain size syllable window at an edge of a word.

Metrical Windows

For a language with (metrical) windows, primary stress may only fall within a certain size syllable window at an edge of a word.

Consider the system of Macedonian (Kager 2012:1457) which has a three-syllable final window.

Metrical Windows

For a language with (metrical) windows, primary stress may only fall within a certain size syllable window at an edge of a word.

Consider the system of Macedonian (Kager 2012:1457) which has a three-syllable final window.

(1) a. ad.vo.ˈkat – 'lawyer'
  b. ad.vo.ˈka.ti – 'lawyers'
  c. ad.vo.ˈka.ti.te – 'the lawyers'

(2) a. kon.zu.**ˈma**.tor – 'consumer'
  b. kon.zu.ma.**ˈto**.ri.te – 'the consumers'

3-Final; Default Antepenultimate

Let's model a language with a three-syllable word-final window where the default position is the penultimate syllable.

3-Final; Default Antepenultimate

Let's model a language with a three-syllable word-final window where the default position is the penultimate syllable.

Descriptively,

- If stress within window, realize faithfully.
- If stress outside window, default to left edge of window (penultimate).

3-Window; Default Antepenultimate

A single distance-scaled constraint cannot generate such a system.

3-Window; Default Antepenultimate

A single distance-scaled constraint cannot generate such a system.

This is because a scaled constraint motivating right alignment of stress would always prefers candidates with stress closest to the right edge.

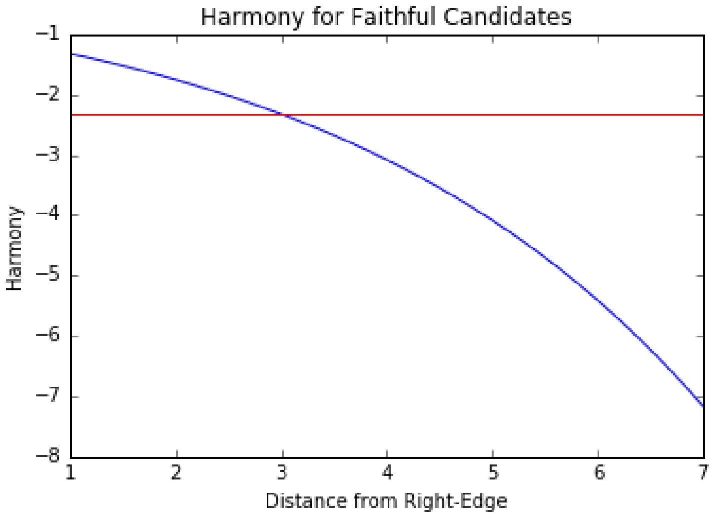
3-Window; Default Antepenultimate

A single distance-scaled constraint cannot generate such a system.

This is because a scaled constraint motivating right alignment of stress would always prefers candidates with stress closest to the right edge.

Changing the scaling factor would allow for different cutoffs of Faith (and thus change the size of the window), but would not create a more optimal candidate anywhere else.

Align-R scaling prevents antepenult default



3-Window; Default Antepenultimate

We want our harmony function to prefer the candidate on the antepenult, while still respecting the metrical window.

3-Window; Default Antepenultimate

We want our harmony function to prefer the candidate on the antepenult, while still respecting the metrical window.

To accomplish this, we introduce a second scaled constraint: Align-L.

3-Window; Default Antepenultimate

We want our harmony function to prefer the candidate on the antepenult, while still respecting the metrical window.

To accomplish this, we introduce a second scaled constraint: Align-L.

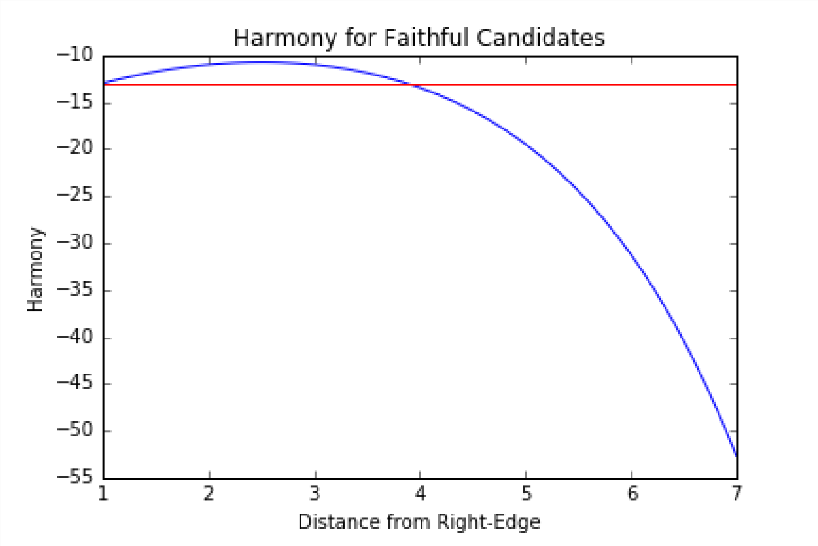
The combination of these two competing forces generates default antepenultimate stress while respecting the window.

3-Final; Default Antepenult with Scaling

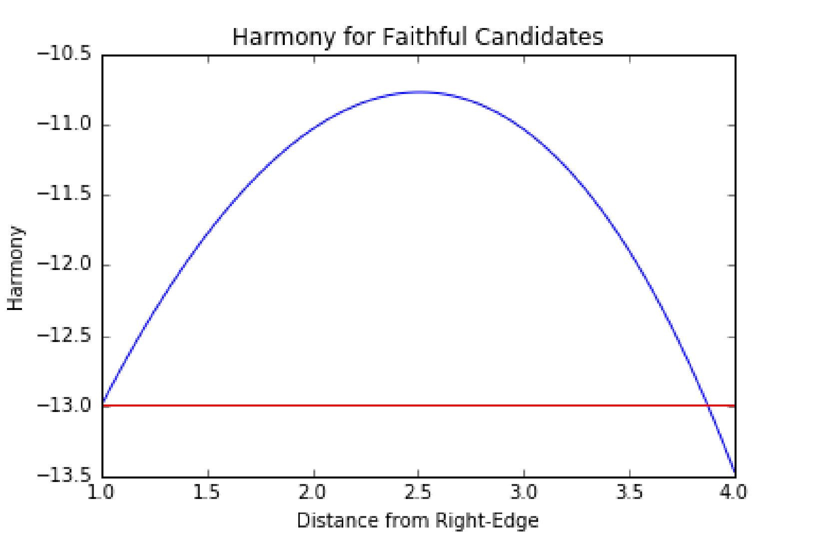
	ssssssS	Faith	Align-R	Align-L
01	→ ssssssS	0	-1*f(1)	-1*g(7)
02	Sssssss	-1	-1*f(7)	-1*g(1)
03	sssssSs	-1	-1*f(2)	-1*g(6)
04	ssssSss	-1	-1*f(3)	-1*g(5)
05	sssSsss	-1	-1*f(4)	-1*g(4)
	ssssSss	Faith	Align-R	Align-L
06	ssssssS	-1	-1*f(1)	-1*g(7)
07	Sssssss	-1	-1*f(7)	-1*g(1)
08	sssssSs	-1	-1*f(2)	-1*g(6)
09	→ ssssSss	0	-1*f(3)	-1*g(5)
10	sssSsss	-1	-1*f(4)	-1*g(4)
	sssSsss	Faith	Align-R	Align-L
11	ssssssS	-1	-1*f(1)	-1*g(7)
12	Sssssss	-1	-1*f(7)	-1*g(1)
13	sssssSs	-1	-1*f(2)	-1*g(6)
14	→ ssssSss	-1	-1*f(3)	-1*g(5)
15	sssSsss	0	-1*f(4)	-1*g(4)

Where $f(n) = \alpha^n$ and $g(n) = \beta^n$

3-Window; Default Antepenultimate



3-Window; Default Antepenultimate



Discussion

- By appealing to both linear and nonlinear distance-based penalty scaling functions, we can capture both bounded and unbounded distance effects
- Distance-based penalty scaling links window phenomena with traditional directional phenomena (see McCarthy 2003's "bounded gradience")
- Beyond the edge effects discussed in this paper, distance-based penalty scaling also can extend to word-internal interactions between segments at a particular distance from one another (see e.g. Zymet 2016)
- Future directions: we believe the approach naturally extends to other instances of window-type effects, such as bounded phonotactic restrictions, or 'trough' effects (see *e.g.* Hyman 1998)

Thank you!

Sharon Inkelas
inkelas@berkeley.edu

Eric Wilbanks
wilbanks_eric@berkeley.edu

Analyses and code available at
https://github.com/EricWilbanks/amp-2017

Solving for Weights/Constant

Now we have to solve for the following values:

- Weight of Faith (x)
- Weight of Align-R (y)
- Scaling constant (α)

Solving for Weights/Constant

Now we have to solve for the following values:

- Weight of Faith (x)
- Weight of Align-R (y)
- Scaling constant (α)

To do this we'll establish an **inequality set** which describes the relationships between winning and losing candidates' harmonies.

Inequality System

	ssssssS	Faith (x)	Align-R (y)
01	→ ssssssS	0	-1*f(1)
02	Ssssss	-1	-1*f(7)
03	sssssSs	-1	-1*f(2)
04	ssssSss	-1	-1*f(3)
05	sssSsss	-1	-1*f(4)

Although we don't yet know the values for the constraint weights or scaling constant, we do know the harmony formula of each candidate.

Inequality System

	sssssssS	Faith (x)	Align-R (y)
01	→ ssssssS	0	-1*f(1)
02	Ssssss	-1	-1*f(7)
03	sssssSs	-1	-1*f(2)
04	ssssSss	-1	-1*f(3)
05	sssSsss	-1	-1*f(4)

Although we don't yet know the values for the constraint weights or scaling constant, we do know the harmony formula of each candidate.

For example, the winning candidate's harmony formula is $0 * x + -1 * y * f(1)$ which reduces to $-y * \alpha^1$. Continuing this for the losing candidates, this tableau can be represented by the following inequality system:

Inequality System

	ssssssS	Faith (x)	Align-R (y)
01	→ ssssssS	0	-1*f(1)
02	Ssssss	-1	-1*f(7)
03	sssssSs	-1	-1*f(2)
04	ssssSss	-1	-1*f(3)
05	sssSsss	-1	-1*f(4)

Although we don't yet know the values for the constraint weights or scaling constant, we do know the harmony formula of each candidate.

For example, the winning candidate's harmony formula is $0 * x + -1 * y * f(1)$ which reduces to $-y * \alpha^1$. Continuing this for the losing candidates, this tableau can be represented by the following inequality system:

$-y * \alpha^1 > -x - y * \alpha^7$
$-y * \alpha^1 > -x - y * \alpha^2$
$-y * \alpha^1 > -x - y * \alpha^3$
$-y * \alpha^1 > -x - y * \alpha^4$

Solving the System

With the inequality system from all input forms in hand, we can attempt to find a solution.

Solving the System

With the inequality system from all input forms in hand, we can attempt to find a solution.

We'll be using non-linear optimization (namely, the sequential least squares programming algorithm as implemented in Scipy) to test for solutions.

Solving the System

With the inequality system from all input forms in hand, we can attempt to find a solution.

We'll be using non-linear optimization (namely, the sequential least squares programming algorithm as implemented in Scipy) to test for solutions.

Our optimization problem is subject to the inequality system we've laid out and the algorithm's goal is to minimize the constraint weights.

Solving the System

With the inequality system from all input forms in hand, we can attempt to find a solution.

We'll be using non-linear optimization (namely, the sequential least squares programming algorithm as implemented in Scipy) to test for solutions.

Our optimization problem is subject to the inequality system we've laid out and the algorithm's goal is to minimize the constraint weights.

For our purposes, we're not interested in the numerical values of the solution, but instead **whether a solution is even possible** given our system.

Selkup Stress Scaling

	/HLHL/	HaveStress	*Ĺ (left-index)	ID (right-index)	
		7	*2*	*1*	$\mathcal{H}$
a.	HLHL	-1			-7
b.	HLHĹ		(-1 *x 4*) = **-4**	(-1 *x 1*) = **-1**	-9
→ c.	HLH́L			(-1 *x 2*) = **-2**	-2
d.	HĹHL		(-1 *x 2*) = **-2**	(-1 *x 3*) = **-3**	-7
e.	H́LHL			(-1 *x 4*) = **-4**	-4

Table 5: Selkup Stress: Heavy and Light Syllables

Selkup Stress Scaling

	/LLLL/	HaveStress	*Ĺ (*left-index*)	ID (*right-index*)	
		7	*2*	*1*	$\mathcal{H}$
a.	LLLL	-1			-7
b.	LLLĹ		(-1 *x 4*) = **-4**	(-1 *x 1*) = **-1**	-9
c.	LLĹL		(-1 *x 3*) = **-3**	(-1 *x 2*) = **-2**	-8
d.	LĹLL		(-1 *x 2*) = **-2**	(-1 *x 3*) = **-3**	-7
→ e.	ĹLLL		(-1 *x 1*) = **-1**	(-1 *x 4*) = **-4**	-6

Table 6: Selkup Stress: Only Light Syllables

Selkup Stress

	hlhl	Have Stress	ID	*ĺ
1	hlhl	-1	0	0
2	→ hlh́l	0	-1	0
3	hĺhl	0	-1	-1
4	h́lhl	0	-1	0
5	hlhĺ	0	-1	-1

	llll	Have Stress	ID	*ĺ
6	llll	-1	0	0
7	lllĺ	0	-1	-1
8	llĺl	0	-1	-1
9	lĺll	0	-1	-1
10	→ ĺlll	0	-1	-1

Selkup Stress NonLinear Scaled

	hlhl	Have Stress (x)	ID (y)	*ĺ (z)
1	hlhl	-1	0	0
2	→ hlh́l	0	-1*f(2)	0
3	hĺhl	0	-1*f(3)	-1*g(2)
4	h́lhl	0	-1*f(4)	0
5	hlhĺ	0	-1*f(1)	-1*g(4)

	llll	Have Stress (x)	ID (y)	*ĺ (z)
6	llll	-1	0	0
7	lllĺ	0	-1*f(1)	-1*g(4)
8	llĺl	0	-1*f(2)	-1*g(3)
9	lĺll	0	-1*f(3)	-1*g(2)
10	→ ĺlll	0	-1*f(4)	-1*g(1)

Where $f(n) = \alpha^n$ and $g(n) = \beta^n$

Selkup Stress Scaled NonLinear Solution

	hlhl	Have Stress (x)	ID (y)	*ĺ (z)	$\mathcal{H}$
1	hlhl	-1	0	0	-2.0000065
2	→ hlh́l	0	-1*f(2)	0	-1.000002
3	hĺhl	0	-1*f(3)	-1*g(2)	-2.00000650001
4	h́lhl	0	-1*f(4)	0	-1.000003
5	hlhĺ	0	-1*f(1)	-1*g(4)	-2.0000045

	llll	Have Stress (x)	ID (y)	*ĺ (z)	$\mathcal{H}$
6	llll	-1	0	0	-2.0000065
7	lllĺ	0	-1*f(1)	-1*g(4)	-2.00000850002
8	llĺl	0	-1*f(2)	-1*g(3)	-2.00000750001
9	lĺll	0	-1*f(3)	-1*g(2)	-2.00000650001
10	→ ĺlll	0	-1*f(4)	-1*g(1)	-2.0000055

Where $f(n) = \alpha^n$, $g(n) = \beta^n$
$x = 2.0000064999999445$, $y = 1.0000009999999973$,
$z = 1.0000009999999957$, $\alpha = 1.0000004999999825$,
$\beta = 1.000001499999813$

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

- Windows may be 2 or 3 syllables long

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

- Windows may be 2 or 3 syllables long
- Windows of each length are attested in both initial and final position

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

- Windows may be 2 or 3 syllables long
- Windows of each length are attested in both initial and final position
- Default position may be sensitive to phonological properties (e.g., "default to right-most heavy syllable in window")

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

- Windows may be 2 or 3 syllables long
- Windows of each length are attested in both initial and final position
- Default position may be sensitive to phonological properties (e.g., "default to right-most heavy syllable in window")

Typology of Metrical Window Systems

The survey of metrical window systems conducted by Kager (2012) found the following typology:

- Windows may be 2 or 3 syllables long
- Windows of each length are attested in both initial and final position
- Default position may be sensitive to phonological properties (e.g., "default to right-most heavy syllable in window")

(35) Default positions in window systems in StressTyp

Final windows	Default = final	Default = penult	Default = antepenult
Two syllables	19	63	
Three syllables	1	19	18

Initial windows	Default = initial	Default = second	Default = third
Two syllables	25	14	
Three syllables	1	0	0

Figure 1: Typology of Metrical Window Systems

3-Final; Default Final

Let's model a language with a three-syllable word-final window where the default position is the final syllable.

3-Final; Default Final

Let's model a language with a three-syllable word-final window where the default position is the final syllable.

Descriptively,

- If stress within window, realize faithfully.
- If stress outside window, default to final.

3-Final; Default Final with Scaling

	ssssssS	Faith	Align-R
01	→ ssssssS	0	-1*f(1)
02	Ssssssss	-1	-1*f(7)
03	sssssSs	-1	-1*f(2)
04	ssssSss	-1	-1*f(3)
05	sssSsss	-1	-1*f(4)

	ssssSss	Faith	Align-R
06	ssssssS	-1	-1*f(1)
07	Ssssssss	-1	-1*f(7)
08	sssssSs	-1	-1*f(2)
09	→ ssssSss	0	-1*f(3)
10	sssSsss	-1	-1*f(4)

	sssSsss	Faith	Align-R
11	→ ssssssS	-1	-1*f(1)
12	Ssssssss	-1	-1*f(7)
13	sssssSs	-1	-1*f(2)
14	ssssSss	-1	-1*f(3)
15	sssSsss	0	-1*f(4)

Where $f(n) = \alpha^n$

3-Window; Default Final Solution

Our system for a 3-syllable metrical window with default final stress is solvable and therefore logically possible.

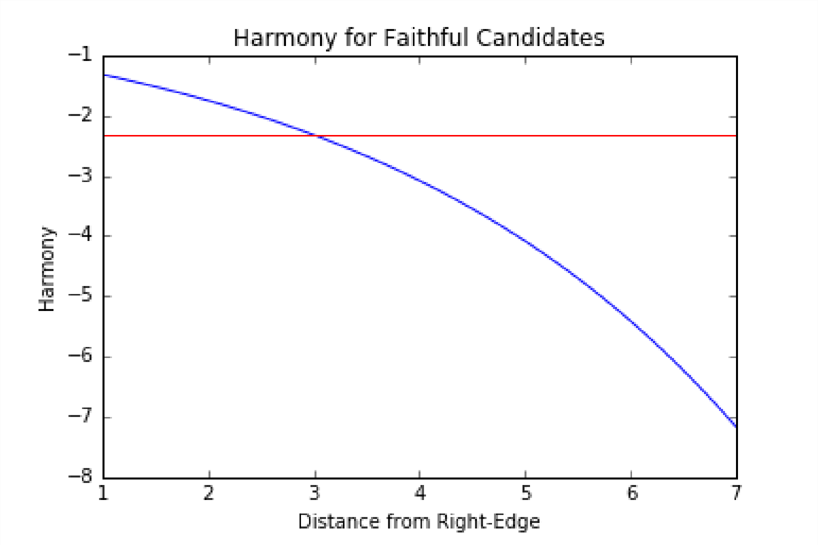
3-Window; Default Final Solution

Our system for a 3-syllable metrical window with default final stress is solvable and therefore logically possible.

$$\begin{aligned} \text{Faith } (x) &= \quad 1.0000009999999984 \\ \text{Align-R } (y) &= \quad 1.0000009999999997 \\ \text{Scaling Factor } (\alpha) &= \quad 1.3247177271169968 \end{aligned}$$

Differences between constraint weights are minuscule, but this is an artifact of our using a minimizing optimizer. There are many other logical systems which are solutions.

3-Window; Default Final



3-Final; Default Final

	ssssssS	Faith	Align-R
1	→ ssssssS	0	-1
2	Sssssss	-1	-1
3	sssssSs	-1	-1
4	ssssSss	-1	-1
5	sssSsss	-1	-1

	sssSsss	Faith	Align-R
11	→ ssssssS	-1	-1
12	Sssssss	-1	-1
13	sssssSs	-1	-1
14	ssssSss	-1	-1
15	sssSsss	0	-1

	ssssSss	Faith	Align-R
1	ssssssS	-1	-1
2	Sssssss	-1	-1
3	sssssSs	-1	-1
4	→ ssssSss	0	-1
5	sssSsss	-1	-1