Convex Regions and Phonological Frequency:
Extending the Weighted Constraints Approach

Olga Dmitrieva
and
Giulio Caviglia

University of California, Berkeley
Purdue University

dmitrieva@berkeley.edu
gcavigli@math.purdue.edu

November 30 2012

Outline

Progression of ambition in the phonological modeling research

- Build models that capture the categorical phenomena: Things that always happen or never happen.
    - Optimality Theory is pretty good at it.
- Build models that capture variation: Things that sometimes happen and sometimes don't.
    - Optimality Theory can be bent to serve this purpose: e.g. Multiple Grammars Theory (Kiparsky 1993, Anttila 1997).
    - Among others are Stochastic OT (Boersma and Hayes 2001) and Noisy Harmonic Grammar (Pater 2009).
- Build models that capture relative frequency of the variable phenomena: Things that happen 70% of the time, and don't happen 30% of the time.
    - There is some exciting research happening (or waiting to happen) here, mostly focusing, for empirical reasons, on within-language variation.
    - Various models with quantitative element arrive at frequency distribution in different ways.

Preview

- In this talk we will propose a method of phonological modeling based on the concept of *weighted* rather than *ranked* constraints.
- Which goes beyond the current use of weights in phonological modeling in a way that seems very natural to us and which can be applied for modeling phonological frequency.
- Along the way, we will raise several "big picture" issues related to the current state of phonological theory: Optimality Theory, Harmonic Grammar, and their relation to each other, in particular.

Big Picture

- Optimality Theory, a currently dominant framework for modeling phonological systems, implements a particular way of selecting the "optimal" output from the set of competing candidates.
- From a purely mathematical perspective, this particular selection method is just one among many and it's choice seems fairly arbitrary.
- The question is: What is so special or particulary natural about this selection mechanism, which makes it appropriate for linguistic modeling? Should we at least consider other options?

Big Picture

- If we choose to embrace weighted constraint approach as an alternative to OT, what are the advantages we will harvest?
  - This question cannot be answered without exploring the formal differences between the two approaches. It is one of the goals of this talk to outline those differences.
- Does the current use of weights in phonological modeling explores the full potential of this approach? What are its limitations and what is a natural pathway for development in this area?
- We will argue that a model proposed here preserves the advantages inherent in the weighted constraints approach and remedies some of the disadvantages present in the current implementation of this method.

Optimality Theory

- Basic architecture: ranked constraints, candidates, violations.
- Output selection process is based on the absolute priority of higher ranked constraints.
- Crucially, in OT, the analysis is only as good as the constraints.

- What evidence can we offer for the validity of the constraints we propose? A couple of options here:
  - "Independently motivated constraints" - constraints that have proven themselves worthy elsewhere.
  - Factorial typology - whether a reasonable typology arises from all possible permutations of your constraints.
- *Factorial typology* is important as the only validity check we have at the moment.

Variation and Frequency

- Although not originally designed for this purpose, an advantage of OT is that it can be used in a fairly straightforward way to model variable phenomena and even their relative frequencies.
- Within-language variation:
  - Relax the assumption that constraints are strictly ranked and establish the rate with which variable outputs win in the resulting grammars.

Input: *t* followed by a vowel

		PARSE-SEG	ONSET	*COMPLEX	ALIGN-L-W
kɔst ʌs	(a) [kɔst][ʌs]		1!	1	
	(b) [kɔs]t[ʌs]	1!	1		
	(c) → [kɔs][tʌs]				1

Predicted output frequencies assuming no rankings

OPTIMAL CANDIDATE	TOTAL RANKINGS	PREDICTED FREQUENCY
[kɔst][ʌs]	5 rankings	21%
[kɔs]t[ʌs]	5 rankings	21%
[kɔs][tʌs]	14 rankings	58%

From Anttila 2012, based on Kiparsky 1993

Variation and Frequency

- Between-language variation:
  - This has already been provided for in standard OT: constraints are freely re-rankable across languages.
  - What about relative frequency of language types? We know that not every possible ranking result in a distinct output pattern (language).
  - If we assume that languages correspond to output patterns, not rankings, every language will be derived by a different number of rankings.
  - Assuming that the probability of each ranking is equal, probability of each language increases as the number of rankings deriving it increases.

Variation and Frequency

- Modeling crosslinguistic vowel inventories (Coetzee 2002)
- Rounding tends to co-occur with backness.
- Constraints: *FrRd, *BkUnRd, Ident(rnd)

Comparison between UPSID, free reranking, and universally fixed ranking

Front round?	Back unround?	Actual	Free reranking	Fixed ranking
No	No	72	33	33
No	Yes	25	16.5	33
Yes	Yes	7	33	33
Yes	No	3	16.5	0

Properties of the Selection Process

- Two abstract properties of the selection process OT relies on are of special importance:
  1. Harmonic Bounding:
     - If candidate's A violations are a proper superset of candidate's B violations, candidate A is harmonically bounded by candidate B: it can never win against it.

(3) Harmonic Bounding. Candidates *a* and *b* are potential winners, *z* is a loser.

C_1>>C_2	C_1	C_2
☞ a		* *
b	*	
z	*	*

C_2>>C_1	C_2	C_1
a	* *	
☞ b		*
z	*	*

From Samek-Lodovici and Prince 1999.

Properties of the Selection Process

- Two abstract properties of the selection process OT relies on are of special importance:
  1. Compatibility with uniform violation addition:
     - Addition of the same violations to each candidate does not change the winning output.

	C_1	C_2
> *a*		**
b	*!	
c	*!	*

	C_1	C_2
> *a*	*	**
b	**!	
c	**!	*

Properties of the Selection Process

- Importantly, many selection processes have these properties. The process used by OT is just one of them.
- Every such selection process corresponds to a monomial order.

'

Monomials, Polynomials and Rings

Using one variable: X (univariate case).

- Monomials: $1, X, X^2, X^3, \ldots$
- Terms: a *term* is a monomial times a number.
  e.g.: $5X$, $7X^5$, $10X^2$.
- Polynomials: a *polynomial* is a finite sum of terms.
  e.g.: $5X + 7X^5 + 10X^2$.
- Polynomial ring: the set of all polynomials.
  It is denoted by $\mathbb{R}[X]$.

Monomials, Polynomials and Rings

Using several variable: $X_1, X_2, \ldots, X_n$ (multivariate case).

- Monomials: a *monomial* is a product of variables.
  e.g.: $X_3, \quad X_1X_2, \quad X_1^2X_5.$
- Terms: a *term* is a monomial times a number.
  e.g.: $5X_1^2X_7^4, \quad X_3^4.$
- Polynomials: a *polynomial* is a finite sum of terms.
  e.g.: $5X_1^2X_7^4 + X_3^4.$
- Polynomial ring: the set of all polynomials in the variables $X_1, \ldots, X_n$.
  It is denoted by $\mathbb{R}[X_1, \ldots, X_n]$.

There are several branches of Mathematics which study *polynomials*, *rings* and other similar structures. Two of such branches are:

- Algebra
- Algebraic Geometry.

Algebra and Algebraic Geometry

The following are two examples of typical (classical) problems:

(A) Find the set of "roots" (solutions) of a given polynomial f in $\mathbb{R}[X]$
(i.e. solve $f = 0$.)
e.g.: $X^2 - 5X + 6 = 0$ solutions: $x = 3$, $x = 2$.

(B) Find the common solutions of a set of polynomials: $f_1, \ldots, f_r$ of $\mathbb{R}[X_1, \ldots, X_n]$.
Study the geometrical properties of such set of solutions.

In general these problems are very hard to solve.
We will focus on (B), keeping in mind that (A) is just a special case of (B).

The typical (*modern*) way to study (B) is:

- Start with $f_1, \dots, f_r$ polynomials in $\mathbb{R}[X_1, \dots, X_n]$.
- Construct an infinite set I of polynomials by using $f_1, \dots, f_r$. This set is called the *ideal generated by* $f_1, \dots, f_r$.
- Find a (possibly) different set $g_1, \dots, g_s$ of polynomials generating I, and with some extra properties. This set is called a Gröbner basis.
- Study properties of $g_1, \dots, g_s$ and (sometimes) properties of an even simpler set consisting of s monomials.

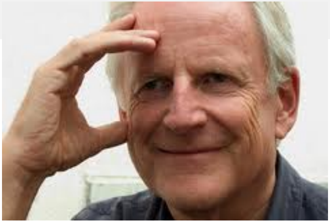
The steps outlined in the previous slide rely (crucially!) on:

- The definition, the existence and the computation of the Gröbner Basis $g_1, \ldots, g_s$.

.... and for them one needs:

- The notion of monomial orders, some theorems and an algorithm due to Bruno Buchberger.

Buchberger's algorithm

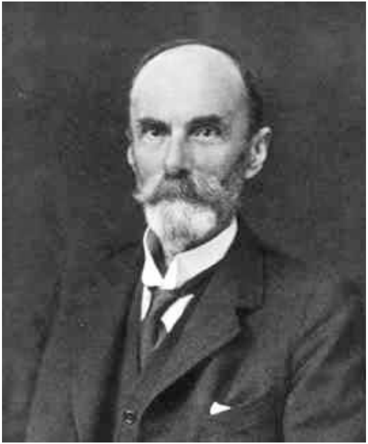


The actual computation of Gröbner bases is possible, in general, thanks to an algorithm discovered by Bruno Buchberger (1965).

"Every set of polynomials can be transformed into a Gröbner basis. This process generalizes three familiar techniques: Gaussian elimination for solving linear systems of equations, the Euclidean algorithm for computing the greatest common divisor of two univariate polynomials, and the Simplex Algorithm for linear programming."
[B. Sturmfels]

Monomial Orders

In 1927 Macaulay was perhaps the first to consider monomial orderings.



He used these orderings to characterize all possible Hilbert functions of graded ideals by comparing them to monomial ideals.

Monomial Orders

We denote a monomial $X_1^{a_1} \cdots X_n^{a_n}$ of $\mathbb{R}[X_1, \ldots, X_n]$ simply by $\mathbf{X}^\mathbf{a}$, where $\mathbf{a} = (a_1, \ldots, a_n)$.
E.g. $X_1^2 X_2 X_3$ in $\mathbb{R}[X_1, X_2, X_3]$ can be written as $\mathbf{X}^\mathbf{a}$ with $\mathbf{a} = (2, 1, 1)$.

A monomial order is a *total order* $<$ on the set of all monomials of R (where *total* means that every two monomials are comparable) such that:

(1) $1 < \mathbf{X}^\mathbf{a}$ for all monomials $\mathbf{X}^\mathbf{a} \neq 1$.

(2) It is *compatible with multiplication*: whenever $\mathbf{X}^\mathbf{a} < \mathbf{X}^\mathbf{b}$ and $\mathbf{X}^\mathbf{c}$ is another monomial, then $\mathbf{X}^\mathbf{a}\mathbf{X}^\mathbf{c} < \mathbf{X}^\mathbf{b}\mathbf{X}^\mathbf{c}$. (equivalently $\mathbf{X}^{\mathbf{a}+\mathbf{c}} < \mathbf{X}^{\mathbf{b}+\mathbf{c}}$.)

These properties, after the correct reinterpretation, are identical to the properties of Harmonic Bounding and Compatibility with uniform violation addition in OT.

Monomial Orders (some examples)

Some examples:

- For $\mathbb{R}[X]$ there is **only one** monomial order, the one induced by the degree: $1 < X < X^2 < \dots$.
- For $\mathbb{R}[X_1, \dots, X_n]$ there are **infinitely many monomial orders** (already with $n = 2$).

  The two most relevant *theoretically* and *computationally* are the lexicographic order (Lex) and the reverse lexicographic order (revLex).

Lexicographic Order (Lex)

Consider $\mathbb{R}[X_1, \ldots, X_n]$.
Think of the variables $X_1, X_2, \ldots, X_n$ as letters of an alphabet where:
X_1 is the *first* letter,
X_2 is the *second* letter,
... and so on...

A monomial is now simply a word.
E.g.: $X_1^3 X_2 X_3^2$ is "$X_1 X_1 X_1 X_2 X_3 X_3$".

The Lexicographic order is obtained by ordering the monomials as words in a dictionary.
For example: $X_1X_1 > X_1X_2 > X_1X_3 > X_2X_2 > X_2X_3 > X_3X_3$.

Reverse Lexicographic Order (RevLex)

Consider $\mathbb{R}[X_1, \dots, X_n]$.
Think of the variables $X_1, X_2, \dots, X_n$ as letters of an alphabet where:
X_n is the *first* letter,
X_{n-1} is the *second* letter,
... and so on...

As before monomial can be view as a word.
E.g.: $X_1^3 X_2 X_3^2$ is "$X_1 X_1 X_1 X_2 X_3 X_3$".

The Reverse Lexicographic order is obtained by ordering the monomials as words in a dictionary (with our new alphabet!), and then reading the dictionary backwards!

For example we have $X_3X_3, X_3X_2, X_3X_1, X_2X_2, X_2X_1, X_1X_1$.
Reading it backwards we obtain: $X_1X_1 < X_1X_2 < X_2X_2 < X_1X_3 < X_2X_3 < X_3X_3$.

Notice: Lex and RevLex looks similar,
for instance they induce the same order on the variables:
$X_1 <_{Lex} X_2 <_{Lex} X_3 <_{lex} \cdots$ and $X_1 <_{RevLex} X_2 <_{RevLex} X_3 <_{RevLex} \cdots$

But they are not the same!
Lex: $X_1^2 < X_1X_2 < X_1X_3 < X_2^2 < X_2X_3 < X_3^2$.
RevLex: $X_1^2 < X_1X_2 < X_2^2 < X_1X_3 < X_2X_3 < X_3^2$.

Initial monomial

Consider $\mathbb{R}[X_1, \ldots, X_n]$ and **fix a monomial order**.
Let $f \in R$ a polynomial.
Write f as sum of monomials with nonzero coefficients.
Denote by $\mathrm{in}(f)$ the first monomial appearing in the summation.
We call it the initial monomial of f.

For instance using RevLex:
let $f = 5X_1X_3 + 4X_2^3 + 3X_1X_3^2$, then $\mathrm{in}_{\mathrm{RevLex}}(f) = X_2^3$.

Computations and Implementations

- Many *invariants* which are important in Commutative Algebra, and Algebraic Geometry can be computed using Gröbner bases. (often the most efficient monomial order for computing such bases is RevLex!).

  Todays Buchberger's algorithm is implemented in many computer-algebra programs, such as: CoCoA, Macaulay2, Magma, Maple, Mathematica, or Singular.

- The pioneering work (in the late 70's) of actually implementing Buchberger's algorithm was done by D.Bayer - M.Stillman and by R. Robbiano.

Use of Gröbner bases in the sciences

"In summary, Gröbner bases and the Buchberger Algorithm for finding them are fundamental notions in algebra. They furnish the engine for more advanced computations in algebraic geometry, such as elimination theory, computing cohomology, resolving singularities, etc. Given that polynomial models are ubiquitous across the sciences and engineering, Gröbner bases have been used by researchers in optimization, coding, robotics, control theory, statistics, molecular biology, and many other fields."
[Sturmfels, American Mathematica Society Notices 2005]

Back to OT: The selection process is done by the RevLex!

	C_4	C_3	C_2	C_1
a	*			*
b		*	*	
☞ c		*		**

- We can associate to each candidate a monomial corresponding to the product (with multiplicity) of the constraints it violates. In the above example a is associated to C_1C_4, b to C_2C_3 and c to $C_1^2C_3$. Moreover we can associate to the Tableau the polynomial: $C_1C_4 + C_2C_3 + C_1^2C_3$.

$$\mathrm{in}_{\mathrm{revLex}}(C_1C_4 + C_2C_3 + C_1^2C_3) = C_1^2C_3.$$

Back to OT:
The selection process is done by the RevLex!

	C_4	C_3	C_2	C_1
a	*			*
b		*	*	
☞ c		*		**

- We can associate to each candidate a monomial corresponding to the product (with multiplicity) of the constraints it violates. In the above example a is associated to C_1C_4, b to C_2C_3 and c to $C_1^2C_3$. Moreover we can associate to the Tableau the polynomial: $C_1C_4 + C_2C_3 + C_1^2C_3$.

$$\mathrm{in}_{\mathrm{revLex}}(C_1C_4 + C_2C_3 + C_1^2C_3) = C_1^2C_3.$$

Monomial orders

- Given that there does not appear to be a strong motivation for using RevLex order to the exclusion of others, why not allow constraints to interact in every possible order?
- Interestingly, a such a model already exists in phonological theory, although in our understanding it has not been used to its full potential: Weighted constraints implemented in Harmonic Grammar.
- Moreover, OT itself was originally inspired by connectionist theory.

Optimality Theory traces its origin to an effort by Prince and Smolensky to combine generative grammar and optimization ideas operative in certain forms of connectionism. (Tesar, Grimshaw, Prince 1999)

Weighted Constraints

The most significant difference between Optimality Theory and connectionist theory is the nature of the harmony function. Connectionist theory uses numerical optimization: the constraints are assigned numeric weights, so the relative strength of different constraints is determined by the relative magnitudes of their respective numeric weights. Optimality Theory uses strict domination optimization (Tesar, Grimshaw, Prince 1999)

The possible relationships between connectionist numeric optimization and Optimality theoretic strict domination optimization is a wide open topic, the subject of much future research. (Tesar, Grimshaw, Prince 1999)

Weighted Constraints

- What is the relationship between numeric weight-based selection and strict domination of OT?
- We claim that from a mathematical point of view the answer to this question is given by a theorem of Robbiano:
  - The choice of the output in OT is determined by the *reverse lexicographic monomial orders*.
  - The choice of the output in the weighted constraints model is determined by *every possible monomial order*.
  - Thus, weighted constraints models can produce the same patterns as the traditional OT and often more.
- More can be done with weights than with standard OT machinery.
- It has inspired some to venture into the weighted constraints territory.

Robbiano's Theorem

Consider $\mathbb{R}[X_1, \ldots X_n]$. Let $\mathbf{w}$ be a vector $(w_1, \ldots, w_n)$ with non-negative entries (i.e. $w_i \geq 0$ for all i)

Every such vector induces a weight function $\mathbf{w}$ from $\mathbb{R}^n \to \mathbb{R}$, $(a_1, \ldots, a_n) \mapsto w_1 a_1 + \cdots + w_n a_n$.

For instance if $\mathbf{w} = (2, 3, 3)$ and $\mathbf{a} = (1, 1, 2)$, then $\mathbf{w}(\mathbf{a}) = 2 \cdot 1 + 3 \cdot 1 + 3 \cdot 2 = 11$.

Every weight function induces a partial order on the monomials of R by setting $\mathbf{X}^{\mathbf{a}} <_{\mathbf{w}} \mathbf{X}^{\mathbf{b}}$ if $\mathbf{w}(\mathbf{a}) < \mathbf{w}(\mathbf{b})$.

Theorem (Robbiano)

For every monomial order $<$ and every finite set of monomials S, there exists a weight $\mathbf{w}$ such that $<$ and $<_{\mathbf{w}}$ agree on S. Moreover every partial order on the monomials of R induced by a weight can be refined to be a monomial order.

Harmonic Grammar

- Constraints are assigned a numerical weight rather than an order.
- Candidate evaluation is based on the sum of the weights of the violated constraints:
  - The candidate with the smallest resulting value wins.
  - Number of violated constraints becomes important.

dog	FAITH [2]	$*\text{VOICE}]_{word}[1]$	
☞dog		*	1
dok	*		2

hund	FAITH [1]	$*\text{VOICE}]_{word}[2]$	
hund		*	2
☞hunt	*		1

Harmonic Grammar

- Avoidance of voiced geminate co-occurrence with other voiced obstruents in loan words in Japanese (from Pater 2009).

/bobu/	IDENT(VC) [1.5]	OCP(VC) [1]	*VC-GEM [1]	
> bobu		*		1
bopu	*			1.5
/webbu/				
> webbu			*	1
weppu	*			1.5
/doggu/				
doggu		*	*	2
> dokku	*			1.5

Variation in Harmonic Grammar

- The variation in loanword emerges from gradual learning (HG-GLA), and from stochastic evaluation (Noisy Exponential HG).

/bobu/		IDENT(VC) [8.63]	OCP(VC) [4.08]	*VC-GEM [4.41]	*Voice [< 0.01]	
> 0.99	bobu		*		**	4.09
< 0.01	bopu	*			*	8.63
/webbu/						
> 0.99	webbu			*	*	4.42
< 0.01	weppu	*				8.63
/doggu/						
0.51	doggu		*	*	**	8.50
0.49	dokku	*			*	8.63

Weighted constraints

- Most of the explorations of the weighted constraints approach focused on finding *a single weight* for each constraint that would derive a desired pattern (Pater, 2009)
  - Establishing whether such weights exist can be done through solving a system of linear inequalities (Potts et al. 2010).
- This essentially amounts to using a different monomial order.

  How can one examine the unbounded space of possible weightings in HG? The answer relates to the fact that even examining the space of possible rankings becomes impractical when constraint sets get large enough (and this happens very quickly, given that number of rankings is the factorial of the number of constraints). (Pater 2009)

Weighted Constraints

- We believe that this approach does not use the full potential of the weighted constraints method:
  - To fully explore the typological implications of the proposed constraints and provide justification for those constraints, in traditional OT, one needs to consider all possible *rankings*.
  - Similarly, to fully explore the implications of weighted constraints, the *space of all possible weights* needs to be considered.
  - An added advantage of this move is that we produce an equivalent of the *factorial typology* in OT, where each possible language corresponds to an area - *a convex region* - in the weight space.

How to Handle the Infinity

- *How* can the unbounded space of possible weights be explored?
  - The weight space can be *normalized*: while it's absolute size will be affected by the normalization, the relative volumes of the convex regions will remain in the same relationships to each other.
  - The weight space can be *sampled*: we do not need to run the algorithm through every possible weight - a sufficiently high number of samples from the normalized weight space will give us a good approximation of the resulting typology.
    - The volume of the convex region in the normalized space can be computed with precision. However, this task is computationally challenging.

Convex Regions and Typological Frequency

- Another bonus of considering the whole weight space is that it has a very natural extension into typological frequency through the relative volumes of the convex regions.
- If we assume that numerical weights for constraints, just as rankings in standard OT, are chosen randomly and with equal probability,
- The probability of landing inside a particular convex regions increases as the volume of the convex region increases.
- Thus, the relative frequency of languages confined to particular convex regions corresponds to the relative volumes of these regions.

Convex Regions and Typological Frequency

- In Pater 2009 analysis the two alternants in the Japanese loan are predicted to occur with a 50-50 probability.
- Convex region method estimates this relationship as a 33-66 probability, with devoicing being the least common option.
- With pattern under consideration being confined to a handful of loanwords it is difficult to evaluate the predictions of the two models.
    - However, Kawahara and Shin-ichiroo Sano report the probability of devoicing in their corpus study at 40%.
    - A value in-between the two predictions, but crucially deviating from the 50-50 distribution in the direction determined by the convex region analysis.

Convex Regions and Typological Frequency

Type	Grammar	Estimated	Inflation	c-volume
1	No geminates	95%	60%	51.5%
2	Intervocalic geminates	2.2%	22%	28%
3	Intervocalic and word-initial geminates	0.9%	10%	13%
4	Intervocalic, word-initial, and preconsonantal geminates	0.1%	4%	5%
5	Intervocalic, word-initial, preconsonantal, and word-final geminates	0.5%	3%	2.6%

- The core of our proposal:
  - Weighted constraints are a desirable alternative to strict domination in OT since this approach removes the arbitrariness of the revlex selection process.
  - When weights are used, the whole weight space, not individually selected weights, should be considered.
- These approach preserves the existing advantages of weighted constraints: The ability to generate patterns unattainable in OT.
- It also provides additional advantages, such as well spelled-out typological predictions of the analysis.
- As well as quantitative predictions, based on the relative volumes of convex regions.
- Moreover, the geometry of the weight-space may give us an additional insight into the way languages operate.
  - For instance, the possibility of variation between certain output types may be related to the relative proximity of the corresponding convex regions in the weight space.

Thank you for your attention!