2008

Gradual Learning and Convergence

Joe Pater

Gradual Learning and Convergence

Joe Pater, University of Amherst, Massachusetts

Boersma and Hayes (2001) present a version of the Gradual Learning Algorithm (GLA; Boersma 1998) that succeeds in learning several cases of phonologically conditioned variation, and serves to model related gradient well-formedness judgments. They argue that this success, along with the robustness of the GLA in the face of noise in the learning data, favors it over the Constraint Demotion Algorithm (CDA; Tesar and Smolensky 1998, 2000), the standard learnability algorithm for Optimality Theory (OT; Prince and Smolensky 1993/2004). Boersma and Hayes (2001: 46) refer to Boersma (1998) for discussion of the issue of whether the GLA shares with the CDA the ability to find a ranking for any set of data that can be captured by an OT grammar, that is, whether it converges on a correct grammar. There we find statements suggesting that the version of the GLA that Boersma and Hayes employ (referred to as the "Maximal GLA" in Boersma 1998) does have this property. The strongest claim is the following:[1]

(1) The algorithms [including the Maximal GLA - *JP*] are ***convergent***: they can learn all OT grammars, from any initial state, without ever getting trapped in local maxima. (p. 274)

A somewhat more cautious assessment comes later:

(2) We have made plausible, though not yet rigorously proved, that the maximal symmetrized gradual learning algorithm is capable of learning any stochastically evaluating OT grammar. (p. 345)

While critiques of the GLA have pointed out that a formal proof of convergence is lacking (Keller and Asudeh 2002, Jäger to appear), they have yet to provide examples of OT grammars that the GLA is unable to learn. As Boersma (2004: fn. 1) and Jäger (to

appear: fn. 1) point out, none of the examples in Keller and Asudeh (2002) involve data that can be described by the stochastic version of OT that Boersma and Hayes assume.

In this squib, I present a simple abstract data pattern that can be captured by an OT ranking, which the version of the GLA in Boersma and Hayes (2001) (henceforth the GLA) consistently fails to find. I then discuss the relationship of the GLA to a learning algorithm that like the GLA gradually adjusts constraint values, but unlike the GLA possesses a convergence/correctness proof. I show that when implemented within Harmonic Grammar (Legendre, Miyata, and Smolensky 1990) this learning algorithm, the Perceptron learner of Rosenblatt (1958), does succeed in learning the data pattern that foils the standard version of the GLA with stochastic OT. I conclude by discussing some directions for further research.

1. The WLW problem

Given a linguistic theory with constraints/parameters/rules of any generality, learners are faced with what Dresher (1999) calls the credit problem. In OT terms, there is usually more than one constraint that favors an optimal form over any one of its competitors, and that can be ranked by the learner over the constraints preferring the competitor.

Such a situation holds in the pair of candidates in (3), illustrated using standard OT notation, and that of Prince (2002a), which I will adopt in the rest of the paper. The optimal form, or Winner, contains a violation of Con2 that is not shared by the competing "Loser", so Con2 prefers the Loser, and assigns an "L". Con1 and Con3, on the other hand, assign unshared violation marks to the Loser, and prefer the Winner.

(3) *OT credit problem schematized*

Input	Con1	Con2	Con3
Winner		*	
Loser	*		*

	W ~ L	Con1	Con2	Con3
Input	$Output_W$ ~ $Output_L$	W	L	W

The correct ranking must place Con1 or Con3 over Con2 – but which one? A decision can be made based on a further piece of data:

(4) *Credit problem resolved*

	W ~ L	Con1	Con2	Con3
In-1	Out-1$_W$ ~ Out-1$_L$	W	L	W
In-2	Out-2$_W$ ~ Out-2$_L$		W	L

The second Winner-Loser pair provides evidence that Con2 dominates Con3, so the first pair must be dealt with by Con1 >> Con2, giving us Con1 >> Con 2 >> Con3.

Unlike the CDA, the GLA does not use this sort of ranking logic to order the constraints. The GLA is cast within a stochastic version of OT that places constraints on a numeric scale (Boersma 1998). Each time the grammar is used to evaluate Input-Output pairings, the numeric values are converted to an OT rank-ordered hierarchy. Learning is on-line and error driven. For each incoming Input-Output pair, the learner takes the Input and finds the optimal Output given the current state of the grammar. If the learner's own Output does not match the observed Output, learning is triggered. With the learners' Output being the Loser, and the observed form the Winner, the GLA adjusts the ranking values as follows (where x is an adjustable value termed "plasticity"):

(5)   Add x to the ranking value of all constraints preferring the Winner, and subtract x from the ranking value of all constraints preferring the Loser

When exposed to a single Winner-Loser pair with the WLW pattern, the GLA will promote both constraints. In the case of the first pair in (4), that means that it promotes both Con1, the constraint that must dominate Con2, as well as Con3, the constraint that the second Winner-Loser pair shows must be at the bottom of the ranking. This is not fatal for the case in (4), however, since Con3 will be pushed down every time the learner encounters the second pair.

However, if we iterate the WLW pattern a few times, we create a learning problem that consistently foils the standard implementation of the GLA:

(6) *The WLW problem*

	W ~ L	Con1	Con2	Con3	Con4	Con5
In-1	Out-1$_W$ ~ Out-1$_L$	W	L	W		
In-2	Out-2$_W$ ~ Out-2$_L$		W	L	W	
In-3	Out-3$_W$ ~ Out-3$_L$			W	L	W
In-4	Out-4$_W$ ~ Out-4$_L$				W	L

This small dataset requires a ranking of considerable depth: Con1 >> Con2 >> Con3 >> Con4 >> Con5. The following set of numerical ranking values get the right result in stochastic OT:

(7)   *Constraint*   *Ranking value*

Con1   100

Con2   85

Con3   70

Con4   55

Con5   40

In the stochastic OT account of variation, the numerical value that maps to a ranking order is not exactly the ranking value, but instead a random sampling from a normal distribution around the ranking value. When two ranking values are close to one another and their corresponding normal distributions overlap sufficiently, their ranking will be seen to vary in repeated samplings. Therefore, to get a (near) categorical outcome, as in our learning problem, the ranking values must be sufficiently far apart. The distances between the constraints in (7) are such that when the grammar was provided with each Input 100,000 times, with the evaluation "noise" setting of 2 (the standard settings for Praat's "get fractions correct"), the correct output was chosen 100% of the time.

I submitted this learning problem to the GLA as implemented in Praat (Boersma and Weenink 2006). For all of the cells marked 'W' in (6), the Loser had one violation mark, and for those marked 'L', the winner had one violation mark.[2] All settings were the default ones, including the "Symmetric All" option, which adjusts the constraint values as in (5). The initial value of the constraints was 100. A typical outcome is shown in (8), alongside a trial that was run with evaluation noise turned off.

(8)	Constraint	Learned Ranking Value (Praat standard settings)	Learned Ranking Value (noise=0)
	Con3	12392.505	6448.611
	Con4	12392.208	6448.611
	Con1	12391.368	6448.610
	Con2	12391.023	6448.610
	Con5	12390.103	6448.610

Ranking values this high appear to indicate non-convergence - the constraints are cycling higher and higher, without settling on a configuration that will consistently choose the observed optima. It is perhaps theoretically possible that the GLA simply hasn't seen enough data, and that it will eventually converge, but even when it is given 1,000,000 pieces of learning data at each of the four plasticity levels, instead of the standard 100,000, it produces a result similar to that in (8), with constraints in the same order, but with much higher values.

The grammar in (8) with "noise" produces variation for all of the inputs. Praat's "get fractions correct" yielded an average percentage correct of 67.3%. This outcome is

typical. Ten separate learning trials, followed by "get fractions correct", produced an average outcome of 67.5% correct, with the highest being 67.8%.

Why does this simple dataset pose such problems for the GLA? A likely source of the difficulty is that Con3 and Con4 assess more Ws than Ls over the Winner-Loser pairs in (6), while Con2, which must be higher ranked than those two, assesses an equal number of Ws and Ls. If errors were made equally across the dataset, Con3 and Con4 would wind up being promoted more quickly than Con2. And in fact, in the results shown for Praat standard settings in (8), and in many replications that I have run, the constraints assessing more W's than L's (Con3, Con4, and Con1) wind up with higher values than those assessing an equal number of W's and L's (Con2 and Con5).

2. The GLA and a Perceptron learner for Harmonic Grammar

As Mark Johnson has pointed out to me, the GLA closely resembles Rosenblatt's (1958) Perceptron, an on-line learner for linear classification (see Collins 2002 for a recent linguistic application, and http://en.wikipedia.org/wiki/Perceptron for an introduction). Its update rule can be stated for the present context as follows:

(9)   Add $n(x-y)$ to the value of every constraint

    Where $0 < n < 1$, x = Loser's violation marks and y = Winner's violation marks

When the Winner and Loser differ by no more than one violation mark on each constraint, the formula in (9) is exactly equivalent to the GLA's update rule, stated in (5). For example, if plasticity in the GLA is set to 0.5, then 0.5 will be added to the value of each constraint that prefers the Winner. Similarly, if the coefficient n is set to 0.5 for the

Perceptron update rule, then the value of a constraint that is violated once by the Loser, but not by the Winner (and thus prefers the Winner) will also be raised by 0.5.[3]

Like the GLA, Perceptron is error-driven, updating its constraint values when the observed Output fails to match the predicted one. However, Perceptron uses a linear model of constraint interaction to calculate the predicted outcome, rather than stochastic OT's rank ordering. In this respect, Perceptron resembles OT's predecessor Harmonic Grammar (HG). This is no coincidence, since Perceptron is a simple single-layer feedforward neural network, and the original version of HG was cast in an elaborated connectionist descendent. An HG linear model of linguistic constraint interaction can be stated as in (10), where C is a vector representing a representation's violation scores on a set of constraints, and W is a vector of coefficients, or weights, for each of the constraints. The well-formedness, or Harmony (H) of a representation (R) is the sum of the weighted violation scores, here notated as the dot product $\langle . , . \rangle$ of the two vectors.

(10) $H(R) = \langle C, W \rangle$

The Harmony score can be used in an OT-like model of grammar to pick the optimal Input-Output mapping from a set of candidates sharing the same Input (as in some versions of HG; see Legendre *et al.* 2006 and other papers in Smolensky and Legendre 2006, as well as Prince and Smolensky 1993/2004, Keller 2000, 2006, Flemming 2001, Prince 2002b, Pater, Bhatt and Potts 2007, Pater, Potts and Bhatt 2007 and Pater to appear). If we translate OT constraint violations to negative integer scores, and weights are non-negative real numbers, the optimal output is the one with the highest Harmony value. The OT-style tableau in (11) illustrates with a simple abstract case. Constraint

weights are shown in the top row, and the Harmony of each candidate Output, calculated as in (10), is shown in the rightmost column.

(11) *A weighted constraint tableau*

Weight	2	1	*H*
Input	Con1	Con2	
☞ Out-1		-1	-1
Out-2	-1		-2

      Praat v. 4.4.02 and subsequent versions offer the option of using HG evaluation, rather than stochastic OT, as the grammar model in learning simulations. In addition, under v. 4.5.21, selection of an HG grammar model automatically changes the learning rule to the one stated in (9) from that of (5), though this is not crucial for the present case, since the distribution of violation marks is such that (5) and (9) are identical. I submitted the data as described in the previous section to Praat v. 4.5.1 with the standard settings for the learner, and with "LinearOT" evaluation, which functions as in (11), selected for its model of grammar ("noise" was also set to zero so as to more closely mimic Perceptron's 'grammar'). After every one of 10 learning trials, "get fractions correct" yielded a success rate of 100%. The same result held when "noise" was included at its standard setting. This success is unsurprising, for two reasons. First, as noted by Karen Jesney (p.c.), this learning problem is much easier in HG than OT, due to additive constraint interaction. With the constraint weights at their initial equal settings of 100, the correct outputs are chosen for the three Input-Output mappings in which two constraints prefer the Winner. The outcome for Input-1 is shown in (12).

(12) *Parsing of Input-1 under initial weighting*

Weight	100	100	100	*H*
In-1	Con1	Con2	Con3	
☞ Out-1$_W$		-1		-100
Out-1$_L$	-1		-1	-200

See Pater *et al.* (2007) and Jesney and Tessier (to appear) for discussion of the consequences of additive interaction and gradual learning for phonological acquisition.

The second reason that the success of Perceptron is unsurprising is that its original formulation is guaranteed to converge on a correct weighting, if it is given sufficient learning data (Novikoff 1962, Collins 2002). This convergence/correctness proof should extend to the current application with HG, at least to a noise-free version.[4]

Further testing of the Perceptron/HG combination on other learning problems does yield a non-obvious result, however: even with noisy evaluation in the grammar model, it appears to continue to converge on weightings that yield correct categorical outcomes (Boersma 2007, Boersma and Pater 2007).

3. Conclusions

The WLW problem is a simple data pattern that the version of the GLA in Boersma and Hayes (2001) fails to learn. This suggests that *pace* Boersma (1998), this version of the GLA is not convergent. For this learning problem, the update rule of the GLA is equivalent to that of the Perceptron model of Rosenblatt (1958), whose original formulation has been proven to converge on a correct system of linear classification. I have shown that when the GLA/Perceptron is run with a linear model of grammar, like that of Harmonic Grammar, it does succeed on the WLW problem. Thus, one reason for

the failure of the Boersma and Hayes (2001) version of the GLA on this problem, and potentially for its failure to provably converge in general, is its use of stochastic OT as a model of grammar.

The GLA is proposed to meet an ambitious set of goals. It aims to learn grammatically conditioned frequency distributions in variation and to model gradual acquisition (Boersma and Levelt 2000) while at the same time converging on correct grammars for categorical patterns. It is worth noting that variant of the GLA's update rule called "Demotion Only" in Praat and the "Minimal GLA" in Boersma (1998) does succeed on the WLW pattern and does have a convergence/correctness proof; see Boersma (1998) for the proof and discussion of issues in learning variation. A further goal is for the algorithm to work with a sufficiently restrictive model of grammar: the GLA adopts OT as a grammatical model for its restrictiveness relative to HG (Boersma 2004: fn. 2, Prince and Smolensky 1993/2004: 232-233).

The consequences of the greater power of HG are the subject of ongoing research (Legendre *et al.* 2006, Pater *et al.* 2007ab, Pater to appear). The importance of this issue is increased insofar as a linear model of grammar is in fact key to a successful gradual theory of learning (see relatedly Wilson 2006 and Jäger to appear; cf. Jarosz 2006, Tessier 2006). A linear or log-linear model of grammar may also be key to a successful theory of variation (see references below) or of other types of grammatical gradience (see e.g. Legendre et al. 1990, Keller 2000, 2006, Keller and Asudeh 2002, Coetzee and Pater 2007, Hayes and Wilson to appear; cf. Boersma 2004). Another open research question is the proper formulation of a grammatical model that copes with variation, and of its

associated learning algorithm. Boersma's (1998) "noise" theory of variation can be incorporated into HG, as in one of the learning simulations reported in section 2. Such a theory remains to be compared with stochastic OT, other theories of variation in OT (e.g. Anttila 1997, Coetzee 2004, Jarosz 2006), and to log-linear models of grammar that directly assign a probability distribution to the candidate set (Johnson 2002, Goldwater and Johnson 2003, Jäger and Rosenbach 2006, Jäger to appear). These theories could be assessed in terms of whether they generate realistic variable and categorical linguistic patterns, and also in terms of the existence and success of related learning algorithms.

Notes

1. This claim is echoed in the Praat OT learning tutorial (published as Boersma 1999: 29; also available in Boersma and Weenink 2006), which states that this version of the GLA is "guaranteed to converge to the target language, if that language can be described by a stochastic OT grammar".

2. The Praat files are available at http://people.umass.edu/pater/WLW-5.txt and http://people.umass.edu/pater/WLW-5dist.txt. Similar results are also obtained with the OT-Soft implementation of the GLA (Bruce Hayes, p.c.).

3. The Perceptron update rule is similar to a number of procedures that adjust coefficient values, including ones in more elaborate connectionist learners, as well as ones in statistical models, most notably stochastic gradient ascent (see relatedly Jäger and Rosenbach 2006, Jäger to appear) and the generalized linear model for regression analysis (see relatedly Keller 2000, 2006). Thanks to Paul Boersma for pointing out these connections.

4. Perceptron is guaranteed to converge on a correct weighting for all cases where correct and incorrect mappings are linearly separable, that is, can be separated by a linear strict inequality. In HG, each optimum is by definition linearly separable from its competitors, and the extension of the Perceptron proof to linguistic categorization in Collins (2002) appears to further extend to the present application. Relatedly, Fischer (2005) provides a convergence proof for a gradual learning algorithm for a log-linear model of probabilistic grammar. Thanks to Paul Boersma, Michael Collins, Gerhard Jäger, Gaja Jarosz, Mark Johnson, Andrew McCallum, and Colin Wilson for discussion.

5. Papers notated as [ROA-xxx] are available at the Rutgers Optimality Archive, http://roa.rutgers.edu/.

References[5]

Anttila, Arto. 1997. Deriving variation from grammar. In Variation, Change and Phonological Theory, ed. Frans Hinskens, Roeland van Hout, and Leo Wetzels, 35- 68. Amsterdam: John Benjamins.

Boersma, Paul. 1998. *Functional phonology: Formalizing the interactions between articulatory and perceptual drives*. Doctoral dissertation, University of Amsterdam.

Boersma, Paul. 1999. Optimality-Theoretic Learning in the Praat Program. In *Proceedings of the Institute of Phonetic Sciences* 23, 17–35. University of Amsterdam. [ROA-380].

Boersma, Paul. 2004. A stochastic OT account of paralinguistic tasks such as grammaticality and prototypicality judgments. Ms., University of Amsterdam. [ROA-648].

Boermsa, Paul. 2007. GLAsuccessRate. Script for Praat software, available at http://www.fon.hum.uva.nl/paul/gla/.

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32: 45-86. [ROA-348].

Boersma, Paul and Clara C. Levelt. 2000. Gradual constraint-ranking learning algorithm predicts acquisition order. In *The proceedings of the thirtieth annual child language research forum*, ed. Eve V. Clark. Stanford: CSLI Publications, 229- 237.

Boersma, Paul, and Joe Pater. 2007. On the convergence properties of a gradual learning algorithm for Harmonic Grammar. Ms, University of Amsterdam and University of Massachusetts, Amherst.

Boersma, Paul and David Weenink. 2006. *Praat: doing phonetics by computer*. Software version 4.5.1. Retrieved October 28, 2006 from www.praat.org.

Coetzee, Andries. 2004. *What it Means to be a Loser: Non-Optimal Candidates in Optimality Theory*. Doctoral dissertation, University of Massachusetts, Amherst.

Coetzee, Andries, and Joe Pater. 2007. Weighted constraints and gradient restrictions on place- co-occurrence in Muna and Arabic. Ms, University of Michigan and University of Massachusetts, Amherst.

Collins, Michael. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 2002.

Dresher, Bezalel Elan. 1999. Charting the Learning Path: Cues to Parameter Setting. Linguistic Inquiry 30: 27-67.

Fischer, Markus. 2005. A Robbins-Monro type learning algorithm for a maximum-entropy maximizing version of stochastic Optimality Theory. Master's thesis, Humboldt University, Berlin. [ROA-767]

Goldwater, Sharon, and Mark Johnson. 2003. Learning OT Constraint Rankings Using a Maximum Entropy Model. In Proceedings of the Workshop on Variation within Optimality Theory, ed. Jennifer Spenader, Anders Eriksson, and Östen Dahl, 111-120. Stockholm University.

Hayes, Bruce and Colin Wilson. To appear. A maximum entropy model of phonotactics and phonotactic learning. In *Linguistic Inquiry*.

Jäger, Gerhard. To appear. Maximum entropy models and Stochastic Optimality Theory, In *Architectures, Rules, and Preferences: A Festschrift for Joan Bresnan*, ed. Jane Grimshaw, Joan Maling, Chris Manning, Jane Simpson, and Annie Zaenen. Stanford, CA: CSLI. [ROA-625].

Jäger, Gerhard, and Anette Rosenbach. 2006. The winner takes it all - almost. Cumulativity in grammatical variation. *Linguistics* 44: 937-971.

Jarosz, Gaja. 2006. *Rich lexicons and restrictive grammars - Maximum likelihood learning in Optimality Theory.* Doctoral dissertation, Johns Hopkins University, Baltimore, Maryland. [ROA-884].

Jesney, Karen and Anne-Michelle Tessier. To appear. Re-evaluating learning biases in Harmonic Grammar. In *University of Massachusetts Occasional Papers* 37, ed. Michael Becker.

Johnson, Mark. 2002. Optimality-theoretic Lexical Functional Grammar. In *The Lexical Basis of Syntactic Processing: Formal, Computational and Experimental Issues*, ed. Suzanne Stevenson and Paola Merlo, 59-73, John Benjamins.

Keller, Frank. 2000. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. Doctoral dissertation, University of Edinburgh.

Keller, F. 2006. Linear Optimality Theory as a Model of Gradience in Grammar. In *Gradience in Grammar: Generative Perspectives*, ed. Gisbert Fanselow, Caroline Féry, Ralph Vogel, and Matthias Schlesewsky. Oxford University Press.

Keller, Frank, and Ash Asudeh. 2002. Probabilistic learning algorithms and Optimality Theory. *Linguistic Inquiry* 33: 225-244. [ROA-675].

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990. Harmonic Grammar – a formal multi-level connectionist theory of linguistic wellformedness: An application. In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 884–891. Cambridge, MA: Lawrence Erlbaum.

Legendre, Géraldine, Antonella Sorace, and Paul Smolensky. 2006. The Optimality Theory–Harmonic Grammar connection. In Smolensky and Legendre 2006, 903–966.

Novikoff, A.B.J. 1962. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata* 12, 615-622. Polytechnic Institute of Brooklyn.

Pater, Joe, Rajesh Bhatt and Christopher Potts. 2007a. Linguistic Optimization. Ms, University of Massachusetts, Amherst. [ROA-924]

Joe Pater, Christopher Potts and Rajesh Bhatt. 2007b. Harmonic Grammar with Linear Programming. Ms, University of Massachusetts, Amherst. [ROA-827]

Pater, Joe. To appear. Review of Smolensky and Legendre 2006. In *Phonology*.

Prince, Alan. 2002a. Arguing Optimality. In *University of Massachusetts Occasional Papers in Linguistics*: *Papers in Optimality Theory II*, ed. Angela Carpenter, Andries Coetzee, Paul de Lacy, 269-304. Amherst, MA: GLSA. [ROA-562].

Prince, Alan. 2002b. Anything Goes. In *New century of phonology and phonological theory*, ed. Takeru Honma, Masao Okazaki, Toshiyuki Tabata, and Shin-ichi Tanaka, 66–90. Tokyo: Kaitakusha. [ROA-536].

Prince, Alan, and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Technical Report, Rutgers University and University of Colorado at Boulder, 1993. Revised version published by Blackwell, 2004. [ROA-537].

Rosenblatt, Frank. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review* 65: 386-408.

Smolensky, Paul, and Geraldine Legendre. 2006. *The harmonic mind: From neural computation to Optimality-Theoretic grammar*. Cambridge, MA: MIT Press.

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29: 229-268.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

Tessier, Anne-Michelle. 2006. *Biases and stages in phonological acquisition*. Doctoral dissertation, University of Massachusetts, Amherst. [ROA-883].

Wilson, Colin. 2006. Learning Phonology with Substantive Bias: An Experimental and Computational Study of Velar Palatalization. *Cognitive Science* 30.5: 945-982.