Learning Competence from Performance Data
Learnability and Simulated Annealing for OT and HG

Tamás Biró

ACLC, University of Amsterdam, The Netherlands

t.s.biro@uva.nl

Amsterdam Center for Language and Communication ACLC

NWO Netherlands Organisation for Scientific Research

ABSTRACT

Given a theory of what grammars consist of, a learning algorithm aims at finding the specific grammar that may have produced the learning data. Grammars are models for the linguistic competence of the speaker and of the hearer. In most approaches, learning data are thought of as being directly produced by the linguistic competence (hence, in the corresponding models, by grammars), which are therefore always grammatical. Alternatively, some random noise can be added to the data, referring in a vague way to speech errors by the speaker, to acoustic distortion and to parsing errors by the hearer. Yet, performance effects can be more complex than mere random noise.

In Optimality Theory, performance is seen as the algorithm implementing the grammar. Smolensky and Legendre (2006) developed a connectionist approach to performance, whereas the approach advocated independently by Bíró (2006) is purely symbolic. Bíró shows that this approach correctly predicts speech error patterns. His Simulated Annealing for Optimality Theory (SA-OT) Algorithm introduces not only *fast speech forms* but also *irregularities*. The latter are local optima that are globally suboptimal, but the algorithm returns them with a high frequency, independently of the cooling schedule. The influence of fast speech forms and of irregularities on learning has been first investigated in Bíró (2007). This poster elaborates on these results by demonstrating how the system's behavior changes if Optimality Theory is replaced by a symbolic Harmonic Grammar. *Paul Boersma*'s update rule is also compared to *Giorgio Magri*'s, while convergence is defined in terms of *Jensen-Shannon divergence*.

COMPETENCE as linguistic optimization

Generative linguistics as an optimization problem: how to map underlying form U onto surface form SF(U)?

$$\mathrm{SF}(U) = \underset{w \in \mathrm{Gen}(U)}{\arg\,\mathrm{opt}}\; H(w)$$

- *Search space* Gen(U): possible forms (candidates).
- *Target function*: "Harmony" $H(w)$. Its range is ranked: $H(w_1) \leq H(w_2)$ or $H(w_2) \leq H(w_1)$.

Introduce *elementary functions* $C_i(w)$ ("constraints" – a misnomer) with a ranked range: $C_i(w_1) \leq C_i(w_2)$ or $C_i(w_2) \leq C_i(w_1)$. Most often, $C_i(w) \in \mathbb{N}_0$. Derive $H(w)$ from these elementary functions:

1. Weighted sum: $H(w) = g_N \cdot C_N(w) + g_{N-1} \cdot C_{N-1}(w) + ... + g_1 \cdot C_1(w)$.
2. OT tableau row: $H(w) = \boxed{C_N(w)}\,\boxed{C_{N-1}(w)}\,\boxed{...}\,\boxed{C_1(w)}$
3. Exponential weights: $H(w) = -C_N(w) \cdot q^N - C_{N-1}(w) \cdot q^{N-1} - ... - C_1(w) \cdot q$

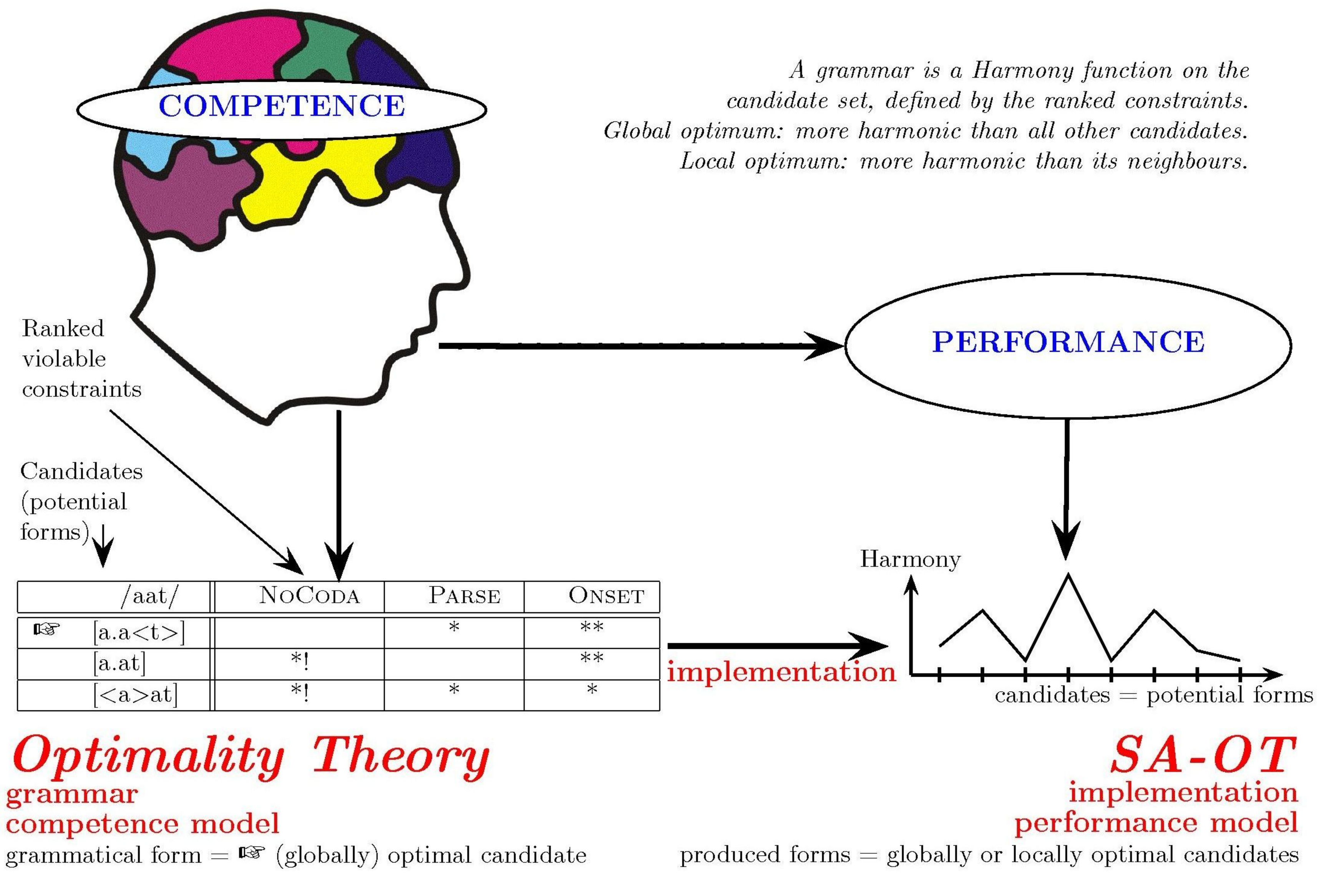
Symbolic OT and HG grammars

N constraints with non-negative integer values. Each constraint C_i has rank $r_i \in \mathbb{R}$. ($r_i \neq r_j$ if $i \neq j$.)
Sort constraints by rank. Place of C_i in this sorted list is $K_i \in \{0, 1, ..., N-1\}$, such that $K_i < K_j$ iff $r_i < r_j$.
q-Harmonic Grammar: $g_i = -q^{K_i}$.
Optimality Theory: lexicographic order; that is, $g_i = -q^{K_i}$ with $q \to +\infty$; that is, $-g_i = \omega^{K_i}$.

String Grammar

A "toy grammar" to be played with, which imitates typical OT phonology:

- *Candidates:* $\mathrm{Gen}(U) = \{0, 1, ..., P-1\}^L$. We have used $L = P = 4$: 0000, 0001, 0120, 0123,... 3333.
- *Neighborhood structure* on this candidate set: w and w' neighbors iff one basic step transforms w to w'.
  Basic step: change exactly one character $\pm 1 \pmod P$ (cyclicity). Each neighbor with equal probability.
  Example: neighbors of 0123 are exactly 1123, 3123, 0023, 0223, 0113, 0133, 0122 and 0120.
- *Constraints* (for all $n \in \{0, 1, ..., P-1\}$):
  - No-n (number of character n in string): $*n(w) := \sum_{i=0}^{L-1}(w_i = n)$.
  - No-initial-n: $*\textsc{Initial}n(w) := (w_0 = n)$.
  - No-final-n: $*\textsc{Final}n(w) := (w_{L-1} = n)$.
  - Assimilation (number of different adjacent character pairs): $\textsc{Assim}(w) := \sum_{i=0}^{L-2}(w_i \neq w_{i+1})$.
  - Dissimilation (number of identical adjacent character pairs): $\textsc{Dissim}(w) := \sum_{i=0}^{L-2}(w_i = w_{i+1})$.
  - Faithfulness to underlying form U (using pointwise distance modulo P):
    $\textsc{Faith}(w) = \sum_{i=0}^{L-1} d(U_i, w_i)$ where $d(a,b) = \min(|(a-b) \bmod P|, |(b-a) \bmod P|)$.



Simulated Annealing

Originating in physics, *simulated annealing* (Boltzmann Machines or stochastic gradient ascent) is a widespread heuristic technique for combinatorial optimization. A random walk is performed on the search space until being trapped in the global or in another local optimum. If target function is real-valued, as in HG, then the slower the speed of the algorithm, the closer to 1 the probability of finding the global optimum. Bíró (2006) demonstrates how to apply simulated annealing in the non-real-valued case of OT, and what its consequences are.

PERFORMANCE or production as implementation

1. *Competence*: the static knowledge	grammatical forms	(explained by) grammar
2. *Mental computation* in the brain	produced forms	implementation of grammar
3. *Performance* in its outmost sense	produced forms	phonetics, pragmatics, etc.

Cf. Bíró (2006:43); Smolensky and Legendre (2006:vol. 1. p. 228). Ways to implement HG and OT:

- **Grammatical:** return the most harmonic candidate (exhaustive search; FS-OT, dynamic programming).
- **Simulated annealing:** return local optima, depending on *cooling schedule* (t_{step}: step by which temperature is decreased in each iteration, "inverse speed").
  - HG: sa converges to gr (frequency of global optimum converges to 1) if $t_{\mathrm{step}} \to 0$ (more iterations).
  - OT: grammatical forms, irregular forms and fast speech forms are returned (Bíró 2007):
    - Grammatical form: globally optimal.
    - Fast speech form: globally not optimal; its frequency converges to 0 if $t_{\mathrm{step}} \to 0$.
    - Irregular form: globally not optimal; its frequency converges to some positive value if $t_{\mathrm{step}} \to 0$.

LEARNING to reproduce teacher's performance

Repeated error-driven updates of the constraint ranks r_i, until convergence:

- Initially: fix random target grammar, fix underlying form, initial random grammar for learner.
- **Error-driven:** "winner" produced by target grammar vs. "loser" produced by learner's current grammar.
- **Update rule:** update the rank r_i of every constraint C_i, depending on whether C_i prefers the winner or the loser. Two approaches ($\epsilon = 0.1$, while ranks are initially random numbers between 0 and $N = 15$):
  - Boersma (1997): increase rank by ϵ if winner-preferring; decrease rank by ϵ if loser-preferring constraint.
  - Magri (2009): increase rank of all winner-preferring constraints by ϵ; decrease rank of highest ranked loser-preferring constraint by $W \cdot \epsilon$, where W is the number of winner-preferring constraints.
- **Convergence criterion:** *JSD* between sample produced by target grammar and sample produced by learner's current grammar $\leq$ average *JSD* of two samples produced by target grammar. (Sample size = 100). Note: we aim at convergence of performance, and not of competence. Child may acquire different grammar.

Jensen-Shannon divergence

A measure of the "distance" of two distributions:

$$JSD(P\|Q) = \frac{D(P\|M) + D(Q\|M)}{2}$$

where $D(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$ (relative entropy, Kullback-Leibler divergence), and $M(x) = \frac{P(x)+Q(x)}{2}$.

- Symmetric: $JSD(P\|Q) = JSD(Q\|P)$. Non-negative: $JSD(P\|Q) \geq 0$. $JSD(P\|Q) \leq 1$.
- $JSD(P\|Q) = 0$ if and only if $P(x) = Q(x),\ \forall x$. $JSD(P\|Q) = 1$ if and only if $P(x) \cdot Q(x) = 0,\ \forall x$.
- Same language: $JSD(L_t\|L_l) = 0$. Not a single overlap: $JSD(L_t\|L_l) = 1$.

Experiment: Measuring number of learning steps

2000 times learning (rnd target, rnd underlying form) per grammar type, production method and learning method.
Distribution of the number of learning steps until convergence: *1st quartile* ; **median** ; *3rd quartile* ; 90th percentile

		OT	10-HG	4-HG	1.5-HG
gramm.	M	*13* ; **27** ; *45* ; 67	*13* ; **28** ; *46* ; 70	*12* ; **27** ; *48* ; 69	*15* ; **30** ; *47* ; 67
	B	*23* ; **43** ; *65* ; 102	*22* ; **41** ; *64* ; 107	*22* ; **42** ; *64* ; 107	*23* ; **40** ; *60* ; 90
sa,	M	*53* ; **109** ; *233* ; 497	*63* ; **140** ; *328* ; 1681	*60* ; **148** ; *366* ; 1517	*83* ; **199** ; *508* ; 1702
$t_{\mathrm{step}} = 0.1$	B	*80* ; **171** ; *462* ; 1543	*92* ; **240** ; *772* ; 7512	*92* ; **239** ; *785* ; 8633	*117* ; **290** ; *694* ; 1956
sa,	M	*64* ; **131** ; *305* ; 1022	*62* ; **134** ; *304* ; 1127	*63* ; **137** ; *329* ; 1278	*72* ; **163** ; *437* ; 2229
$t_{\mathrm{step}} = 1$	B	*90* ; **212** ; *560* ; 1966	*92* ; **233** ; *572* ; 3116	*84* ; **212** ; *646* ; 3005	*101* ; **242** ; *616* ; 2091

CONCLUSION, FUTURE WORK

Observations: from these preliminary experiments (significance based on Wilcoxon rank-sum test):

- Generally, errors make grammars more difficult to learn:
  Production = grammatical *easier than* Production = 0.1-sa *easier than* Production = 1-sa.
- But it seems that for HG
  Production = 1-sa *easier than* Production = 0.1-sa (either significant, or not significant tendency).
- Magri's update rule (M) quicker than Boersma's (B) (extremely significant). Due to larger update steps?
- Grammar type (OT, q-HG): only minor influence ("hardly any" and "small, but very significant").
  OT much easier to learn than 1.5-HG (significant difference for sa cases). NB: also quicker to produce.

Future work:

- Error analysis, source of difficulty: target grammar, learner's initial grammar or learning data order?
- Effect of fast speech forms vs. irregular forms. Production errors made by teacher vs. by learner.
- New update rules, based on the heuristic that produced forms must be local optima.

References

Tamás Bíró (2006). *Finding the Right Words: Implementing Optimality Theory with Simulated Annealing.* PhD thesis, University of Groningen. Also as ROA-896.

Tamás Bíró (2007). 'The benefits of errors: Learning an OT grammar with a structured candidate set'. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 81–88, ACL Prague, June 2007. ROA-929.

Paul Boersma (1997). 'How we learn variation, optionality, and probability'. IFA Proceedings 21: 43-58.

Giorgio Magri (2009). 'New update rules for on-line algorithms for the Ranking problem in Optimality Theory'. Handout. LMA workshop, DGfS 31, Osnabrück, March 2009.

Paul Smolensky and Géraldine Legendre (eds.) (2006). *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar.* MIT Press, Cambridge.