Maximum Entropy Models and Stochastic Optimality Theory

Gerhard Jäger

jaeger@ling.uni-potsdam.de

1 Introduction

In a series of recent publications (most notably Boersma (1998); see also Boersma and Hayes (2001)), Paul Boersma has developed a stochastic generalization of standard Optimality Theory in the sense of Prince and Smolensky (1993). While a classical OT grammar maps a set of candidates to its optimal element (or elements), in Boersma's Stochastic Optimality Theory (StOT for short) a grammar defines a probability distribution over such a set. Boersma also developed a natural learning algorithm, the **Gradual Learning Algorithm** (GLA) that induces a StOT grammar from a corpus. StOT is able to cope with natural language phenomena like ambiguity, optionality, and gradient grammaticality, that are notoriously problematic for standard OT.

Keller and Asudeh (2002) raise several criticisms against StOT in general and the GLA in particular. Partially as a reaction to that, Goldwater and Johnson (2003) point out that maximum entropy (ME) models, that are widely used in computational linguistics, might be an alternative to StOT. ME models are similar enough to StOT to make it possible that empirical results reached in the former model can be transferred to the latter, and these models have arguably better formal properties than StOT. On the other hand, the GLA has a higher cognitive plausibility (as can be seen from Boersma and Levelt (2000)) than the standard learning algorithms for ME models.

In this paper I will argue that it is possible to combine the advantages of StOT with the ME model. It can be shown that the GLA can be adapted to ME models almost without modifications. Put differently, it turns out that the GLA is the single most natural on-line learning algorithm for ME models. Keller and Asudeh's criticism, to the degree that it is justified, does not apply to the combination of ME evaluation with GLA learning, and the cognitive advantages of the GLA are maintained.

2 Stochastic OT and the Gradual Learning Algorithm

I will give only a brief sketch of Boersma's model here — the interested reader is referred to Boersma and Hayes (2001) for a more thorough introduction.

StOT shares the generator component with standard OT. It also uses a set of ranked and violable constraints as the basis for grammatical evaluation. The constraints are not ranked on an ordinal scale though, but each is assigned a real number on a continuous scale, its *rank*. This way it is possible to speak of the distance between two constraints in a meaningful way. In each evaluation event, some random noise is added to each constraint rank. The ranking of a constraint after adding the noise is called the *selection point*. The constraints can be ordered according to the value of their selection points, and this ordering can be used as ranking in the standard OT sense. However, adding the noise value to the ranks of the constraint may change their ordering, so the ranking of selection points may differ from evaluation to evaluation. In this way the ranking on the continuous scale defines a probability distribution over ordinal rankings. This in turn defines a probability distribution over the set of candidates — the probability of a candidate to be optimal is the sum of the probabilities of all ordinal rankings that make it optimal.

The noise that is added to each constraint rank is a normally distributed random variable with mean 0 and standard deviation 1. The probability of a constraint ranking $C_1 > C_2 > \cdots > C_n$ can thus be given by the following formula (where r_i is the rank of constraint C_i and N is the standard normal distribution):

$$P(C_1 > \cdots > C_n) = \int_{-\infty}^{+\infty} dx_1 N(x_1 - r_1) \int_{-\infty}^{x_1} dx_2 N(x_2 - r_2) \cdots \int_{-\infty}^{x_{n-1}} dx_n N(x_n - r_n)$$

A StOT grammar adequately describes a language if it assigns probabilities to the linguistic signs (sentences, syllable structures or whatever) in the corpus that match with their empirical relative frequencies in this language.

If the distances between the ranks of the constraints are very high, the probability of the ordinal ranking that matches the ordering of the ranks converges towards 1. Standard OT, where there is only one ranking, can thus be seen as a borderline case of StOT.

The GLA is an algorithm that takes an analyzed corpus as input and returns a rank for each constraint. Learning is successful if the acquired ranking adequately describes the language from which the corpus was drawn. The generator and the constraint set are known by the learner in advance. By analyzed corpus I mean that both the input and the output (underlying form and surface form for production and conversely for comprehension) are visible to the learner. Likewise, the degree of violations that each constraint incurs on each observation are accessible to the GLA. Each observation thus assigns a natural number to each constraint (the number of stars, so to say), and this is the only information that the GLA needs.

Each constraint is a function from candidates to natural numbers (the number of violations it incurs). I'll write these functions as c_1, c_2 etc. to distinguish the function from the name of the constraints. A ranking is a vector of real numbers (the ranks of the constraints). Observations — input-output pairs — are fed to the GLA in a sequence $(i_1, o_1), (i_2, o_2), (i_3, o_3), \ldots$ of independent trials. At each point in the learning process, the learner assumes some hypothetical ranking vector $\vec{r}$ that is possibly adjusted in each step. A ranking vector defines a probability distributions over output candidates. For each observed output o_n, the learner hypothesizes an output h_n for the input i_n that is drawn according to the probability distribution that corresponds to $\vec{r}_n$. (i_n, h_n) is compared to the observation (i_n, o_n), and the adjustment of $\vec{r}$ depends on this comparison. All constraints that favor the observation over the hypothesis receive a higher rank and vice versa. There is a positive constant η, the learning rate or *plastic-*

ity, that determines the impact of a single learning event. So the update rule for the GLA is

$$(\vec{r}_{n+1})_i = (\vec{r}_n)_i + \eta \cdot \mathrm{sgn}(c_i(i_n, h_n) - c_i(i_n, o_n))$$

(The sign operator "sgn" maps positive numbers to 1, negative numbers to -1, and 0 to itself.) The initial grammar of the learner, $\vec{r}_1$, has to be given *a priori*. If we simplify things further by stipulating that constraints are binary, the sign operator can be omitted.

This formula involves two random variables that are drawn from different distributions. (i_n, o_n) is drawn according to the empirical probability distribution p_{emp}. The probability of a learner's hypothesis depends on two independent distributions — the empirical probability of an input follows from the empirical distribution over input-output pairs and the structure of **GEN**:

$$p_{emp}(i) = \sum_{o:\mathbf{GEN}(i,o)} p_{emp}(i, o)$$

Each $\vec{r}$ defines a probability distribution over the possible outputs for a given input, written as the conditional probability $p_{\vec{r}}(\cdot|i)$.

(i_n, h_n) is drawn from the product of these two distribution $p_{\vec{r}_n}$. To keep notation simple, I'll use the convention

$$p_{\vec{r}}(i, o) = p_{emp}(i) \cdot p_{\vec{r}}(o|i)$$

The expectation values under the distributions p_{emp} and $p_{\vec{r}}$ are written as E_{emp} and $E_{\vec{r}}$ respectively. As long as all constraints are binary, the expected adjustments of the constraints is then

$$(\vec{r}_{n+1})_i \quad = \quad (\vec{r}_n)_i + \eta \cdot (E_{\vec{r}_n}[c_i] - E_{emp}[c_i]) \tag{1}$$

So in effect the GLA compares the average degree of violations of each constraint in the training corpus with its expected degree of violations according to the current grammar of the learner. The algorithm converges if these two values coincide for each constraint.

There is, however, no general proof that the GLA will always converge towards a correct constraint ranking, even if the training corpus is generated by some StOT grammar, and the correct set of constraints is available to the algorithm.[1] This is a serious drawback, and it seems worthwhile to consider alternative models of probabilistic generalizations of OT.

[1] Keller and Asudeh (2002) criticize several aspects of StOT and the GLA. The fact that there is no proof of convergence is the only serious one, I think. They give some examples for languages that the GLA is unable to learn, but these cases involve harmonically bounded candidates with a non-zero probability. It is a general feature of OT that harmonically bounded candidates are always ungrammatical, and it is thus no defect of the GLA that it is unable to learn languages which behave otherwise. Also, I do not think that testing the GLA on unseen data would reveal much about the GLA as such. Testing on unseen data is important to detect over-training. Whether or not training a StOT grammar with some corpus using the GLA leads to over-training depends mainly on the set of constraints and the size of the training corpus; the training algorithm as such has little bearing on that.

3 The maximum entropy approach

Goldwater and Johnson (2003) compare StOT with Maximum Entropy models (or, as they are sometimes called, log-linear models) that are state of the art by now in computational linguistics (see for instance Berger, Della Pietra, and Della Pietra (1996) or Abney (1997)). Let me briefly explain what "maximum entropy" means.

Suppose we know that a certain experiment has two possible outcomes, A and B, but we don't know anything else about it. Which probability should we assign to A and B? The best answer seems to be: 50% probability for each. Likewise, if there are five possible outcomes, A, B, C, D and E, the best estimate is to assign 20% to each if we don't have further information. Every other distribution of the probability mass would represent a bias which is not supported by knowledge. And if we also know that the outcome will be A or B with a probability of 70%? Then the least biased estimate is to assign 35% to both A and B, and 10% to each of the three other events. There is a clear intuition that the least biased hypothesis is the most parsimonious one.

The information theoretic notion of *entropy* quantifies the bias of a probability distribution. The entropy H of a probability distribution p is defined as

$$H(p) = \sum_x p(x) \log \frac{1}{p(x)}$$

The least biased distribution has the highest entropy, and vice versa. If we have partial knowledge about a stochastic process and we have to estimate the underlying probability distribution, the best guess is to choose among all distributions that are compatible with our knowledge the one with the highest entropy.

Let us assume that the unknown probability distribution is a language L in the StOT sense, i.e. a probability distribution over a set of input-output pairs. We know the set **GEN** of possible elements of the language (the generator) and a set of constraints. We also know how many violations each constraint incurs on each candidate, the marginal probabilities of the different inputs, and — crucially — we know how often each constraint is on average violated in the language in question. This may be the result of investigating a large sample of L, but the only empirical facts we are able to observe are the inputs and the number of constraint violations of each observation. So we are looking for a relative probability distribution over the potential output for each input which predicts the correct average degree of violation of each constraint, and among all distributions with this property, we will choose the one with the highest entropy. It can be shown (see Della Pietra, Della Pietra, and Lafferty (1995) for a proof) that this distribution takes the following form:

$$p_{\vec{r}}(o|i) = \frac{1}{Z_{\vec{r}}(i)} \exp(\sum_j r_j c_j(i,o))$$

where $Z_{\vec{r}}(i)$ is a normalization constant which ensures that the probabilities of all candidates sum up to 1. It holds that

$$Z_{\vec{r}}(i) = \sum_{o:\mathbf{GEN}(i,o)} \exp(\sum_j r_j c_j(i,o))$$

Taking the logarithm on both sides yields

$$\log p_{\vec{r}}(o|i) = \sum_j r_j c_j(i,o) - \log Z_{\vec{r}}(i)$$

So the logarithm of the probability of a candidate is a linear function of its constraint violations. (Therefore these probability distributions are called "log-linear".) Of course there are infinitely many log-linear distributions, depending on the values of the rank parameters r_j. Della Pietra, Della Pietra, and Lafferty (1995) also show that among all these log-linear distributions, the one which maximizes the likelihood of the language L is the one which assigns the correct average degree of violations to each constraint. In other words, the unique log-linear distribution which assigns maximal likelihood to L is at the same time the unique distribution with the empirically correct predictions of average constraint violations that maximizes entropy.

There are several learning algorithms available for ME models. Della Pietra, Della Pietra, and Lafferty (1995) developed *improved iterative scaling* specifically for this application. Besides, standard machine learning algorithms like *gradient ascent* or *conjugate gradient ascent* are applicable as well and guaranteed to converge towards the correct distribution.

The general setup of a maximum entropy model is very similar to a StOT grammar. The main difference between StOT and ME is the evaluation component, i.e. the way in which constraint ranks are interpreted as a probability distribution. Like StOT, ME models can be seen as a generalization of standard OT. (In the ME context, constraint violations have to be interpreted as negative numbers in the ME context and as positive integers in StOT to facilitate comparison. I will tacitly do this from now on.) If the ranks (or "weights", as the parameters $\vec{r}$ are usually called in the ME tradition) of the constraints are very high and spread far apart, the probabilities of candidates that would be sub-optimal in classical OT converge towards 0 in the ME interpretation.

The models differ slightly in their treatment of cumulativity of constraint violations. Both StOT and ME diverge from classical OT in admitting *ganging-up effects*. By this I mean that violations of several lower-ranked constraints may jointly outweigh a violation of a higher ranked constraint. ME furthermore predicts *counting cumulativity*. This means that a higher degree of violations of one constraint always reduces probability, no matter what other constraints and other candidates do. In StOT only the relative strength of a constraint violation of a candidate in comparison with other candidates matters. In the present paper I will restrict attention to binary constraints, and thus such effects are excluded by definition.

Goldwater and Johnson (2003) repeat all experiments from Boersma and Hayes (2001) but use ME and conjugate gradient ascent instead of StOT and GLA. They show that the results of both approaches are very similar. So for all we know, there are no clear empirical reasons to favor the one model over the other. From a theoretical point of view ME seems to be superior though because the available learning algorithms are guaranteed to converge.

On the other hand, Paul Boersma (personal communication) mentions two advantages of StOT/GLA over the standard learning algorithms for ME models. First, all these algorithms are off-line algorithms, while the GLA works on-line. This means that improved iterative scaling or (conjugate) gradient ascent needs to "know" the average degree of violation of each constraint *before* the actual learning starts. In practice this means that first the training corpus has to be analyzed in its entirety before learning can take place. The GLA, on the other hand, works gradual. It processes the observations one at a time and gradually approaches the target distribution. Human language acquisition obviously works on-line. Therefore the GLA is *prima facie* cognitively more plausible. Furthermore, the study Boersma and Levelt (2000) demonstrates for a certain empirical domain that the GLA actually predicts the correct order of acquisition. An off-line algorithm is by its very nature unable to make predictions about acquisition order.

4 GLA = Stochastic Gradient Ascent

One of the standard machine learning algorithms that is applicable to ME models is *Gradient Ascent*. This algorithm finds a local maximum of a real-valued function f in an n-dimensional Euclidian space provided f is continuous and differentiable. For ME learning, we have to find the ranks that maximize the likelihood of the training corpus. Since the logarithm is a monotonically increasing function, we can as well maximize the average log-likelihood, i.e. we have to find the parameters $\vec{r}$ that maximize the empirical average of the log-likelihood according to $p_{\vec{r}}$. It is given by the formula

$$f(\vec{r}) = E_{emp}[\log p_{\vec{r}}(i,o)]$$

which is a continuous and differentiable function of the rank vector $\vec{r}$. The log-likelihood has the desirable property of being convex, which means that it does not have local maxima. Gradient Ascent is thus guaranteed to find the global maximum. The gradient of a function f at a point $\vec{r}$ is the vector pointing from $\vec{r}$ in the direction where f grows steepest. Gradient Ascent is an iterative algorithm that, starting from some initial point $\vec{r}_1$, moves small steps in the direction of the local gradient. Since each step increases the value of f, this process is bound to approach the global maximum.[2]

The jth component of the gradient of a function $g(\vec{r})$ is the partial derivative $\frac{\partial}{\partial r_j} g(\vec{r})$. For log-likelihood in a ME model it holds that

$$\frac{\partial}{\partial r_j} f(\vec{r}) \quad = \quad E_{emp}[c_j] - E_{\vec{r}}[c_j] \tag{2}$$

Note that this gradient is exactly the direction of the update in the GLA (1), modulo the convention that constraint functions take non-negative values in StOT and non-positive values here. The update rule for gradient ascent is thus identical to the expected effect of the GLA update rule.

Gradient ascent is still an off-line algorithm though. The expectation values of the constraints have to be known in advance. However, it can easily be transformed into an on-line algorithm. The on-line version is known as *Stochastic Gradient Ascent* (SGA for short) and is a standard technique for training connectionist networks. Here one observation is processed at a time, and the

[2] Provided the function is convex. In the general case only convergence towards a local maximum is guaranteed.

empirical expectation value of each constraint is replaced by the observed value. Adapted to ME, the update rule thus is

$$(\vec{r}_{k+1})_j = (\vec{r}_k)_j + \eta \cdot (c_j(i_k, o_k) - E_{\vec{r}}[c_j])$$

Recall that the distribution $p_{\vec{r}}$ is a product of the theoretical conditional probability distribution $p_{\vec{r}}(o|i)$ and the marginal empirical probabilities of the inputs $p_{emp}(i)$. The latter cannot be calculated analytically. Therefore $E_{\vec{r}}[c_j]$ has to be estimated as well. We thus replace it by a random variable that has $E_{\vec{r}}[c_j]$ as its expected value. This can be done by drawing a sample h_k from the distribution $p_{\vec{r}}(\cdot|i)$ corresponding to the current rankings $\vec{r}$. This leads us to the revised update rule

$$(\vec{r}_{k+1})_j = (\vec{r}_k)_j + \eta \cdot (c_j(i_k, o_k) - c_j(i_k, h_k))$$

For the special case that all constraints are binary, this *is* the Gradual Learning Algorithm, and for the general case it is very closely related to it.

Let us consider the consequences of this finding. Both the GLA for StOT and SGA for ME are on-line algorithms. They both converge iff the empirical and the acquired expectation value for each constraint coincide, i.e. if for all constraints C_j:

$$E_{emp}[c_j] - E_{\vec{r}}[c_j] = 0$$

Under the ME interpretation, among all distributions sharing this property, the learning algorithm finds the one with the highest entropy. If the GLA under the StOT interpretation converges towards a different distribution for the same training corpus, it will be one with lower entropy, i.e. a less parsimonious hypothesis.

5 Order of acquisition

Boersma and Levelt (2000) report an experiment where the acquisition of Dutch syllable structures was simulated using StOT and the GLA. I repeated the experiment with a ME model and SGA and obtained almost identical results.

The attested frequency of Dutch syllable types in child directed speech is as in table 1 (taken from Boersma and Levelt (2000) who attribute it to Joost van de Weijer, personal communication).

CV	44.81%	CCVC	1.98%
CVC	32.05%	CCV	1.38%
VC	11.99%	VCC	0.42%
V	3.85%	CCVCC	0.26%
CVCC	3.25%		

Tab. 1: Frequency of syllable types in Dutch

Following the optimality theoretic acquisition study Levelt and van de Vijver (1998), Boersma and Levelt (2000) assume the following five constraints pertaining to syllable structure:

- *Coda: “don’t produce codas” (violated by -C and -CC syllables)
- Onset: “don’t produce vowel-initial syllables” (violated by V- syllables)
- *ComplexCoda: “don’t produce complex codas” (violated by -CC syllables)
- *ComplexOnset: “don’t produce complex onsets” (violated by CC-syllables)
- Faith: “don’t add or delete material”

The set-up is speaker oriented. Both input and output are assumed to be (or to specify) syllables structures. The first four constraints are markedness constraints and only apply to the output. The last constraint is violated whenever input and output differ.

Boersma and Levelt’s experimental setup is as follows. Each of the nine syllable types can be both input and output. There are thus 81 potential input-output pairs. Faith is never violated in the adult language — in the training corpus input and output always coincide. The corpus frequencies given above are taken to represent the empirical probabilities.

In the initial state Faith is ranked low and the markedness constraints high (a standard assumption in the OT literature). The GLA gradually lowered the markedness constraints and simultaneously raised Faith until it became the strongest constraint. The algorithm converges towards the following ranking:

$$\text{Faith} \gg \text{*ComplexOnset} \gg \text{*ComplexCoda} \gg \text{Onset} \gg \text{*Coda}$$

The final relative ranking of the markedness constraints emerges early during the learning process. Faith outranks the markedness constraints thus not all at once but one by one, starting with *Coda and ending with *ComplexOnset. The intermediate stages correspond to rankings where the learner would produce some, but not all syllable types faithfully. Furthermore, during the phases where Faith and some markedness constraint have similar ranks, the learner would assign different outputs for the same input substantial probabilities, i.e. the learner is predicted to pass stages of optionality before complex syllable types are firmly acquired. This effect does in fact occur during human language acquisition.

In Boersma and Levelt’s simulations, the GLA learned the Dutch syllable types most of the time in the following order:

$$\text{CV} < \text{CVC} < \text{VC} < \text{CVCC} < \text{CCVC} < \text{CCVCC}$$

In about 30% of the experiments, the order of CCVC and CVCC was reversed. The author report no results on the other three syllable types.

According to Levelt and van de Vijver (1998), Dutch children acquire the syllable types in one of the following two orders, where types enclosed in curly brackets are acquired more or less simultaneously.

- CV < CVC < V < VC < {CVCC, VCC} < {CCV, CCVC} < CCVCC
- CV < CVC < V < VC < {CCV, CCVC} < {CVCC, VCC} < CCVCC

The simulation results do in fact fit rather well with the empirical acquisition data.

I repeated Boersma and Levelt's experiment using a ME model and Stochastic Gradient Ascent (SGA). I used a learning rate $\eta = 0.1$. In the initial state, FAITH had a rank of 0.0, and all other constraints a rank of 10.0. The constraint rankings as a function of time is given in figure 1.

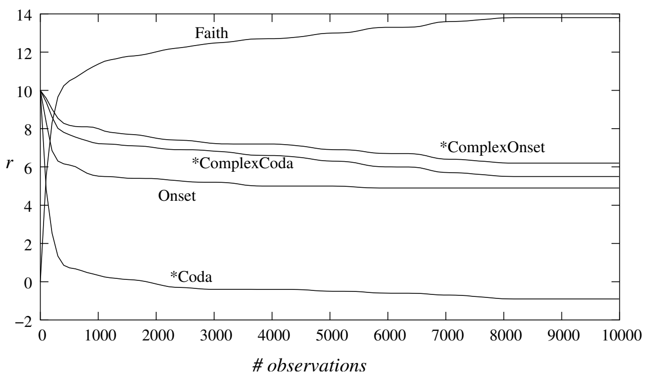

Fig. 1: Learning of constraint ranks

The ME learning algorithm acquires the same constraint ranking as the GLA, and it does so with a qualitatively similar trajectory. This fact, as well as the order of acquisition to be reported shortly does not depend on the learning rate and the precise initial ranking of the markedness constraints as long as they are initially ranked above FAITH and the latter is initially ranked at 0.

Since the probability of a candidate is a continuous function of the constraint rankings both in Boersma's and in the ME model, the acquisition of syllable types is predicted to be gradual in both theories.

In the experiments I conducted, the syllable types where consistently acquired in the following order:

$$\text{CV} < \text{CVC} < \text{V} < \text{VC} < \text{CVCC} < \{\text{CCV, CCVC}\} < \text{VCC} < \text{CCVCC}$$

This is very close to the empirically observed acquisition orders, except that CVCC is acquired too early or VCC too late. It is also consistent with Boersma and Levelt's results. From the available information it is hard to tell whether or not their results are closer to the empirical data than mine, but the difference is certainly small.

6 Conclusion

In this article I compared two stochastic generalizations of Optimality Theory, namely Paul Boersma's Stochastic Optimality Theory and Maximum Entropy models that are frequently used in machine learning contexts. As pointed out by Goldwater and Johnson (2003), the empirical predictions of the two models are rather similar. The latter is advantageous from a formal point of view because there exist provably correct learning algorithms for Maximum Entropy models while the learnability problem for Stochastic OT is still open. On the other hand, Boersma's Gradual Learning Algorithm for Stochastic OT has been successfully applied in many contexts, and it is arguably a better model of human language acquisition than the off-line training algorithms for ME models that are standardly used.

I demonstrated that the Gradual Learning Algorithm is not confined to StOT. It can be applied to ME models as well. In this context it is actually a variant of the Stochastic Gradient Ascent training algorithm that is well-established for other machine learning problems. The insights about language acquisition that the GLA offers are maintained if it is combined with the Maximum Entropy approach to constraint evaluation.

References

Abney, Steven. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23:597–618.

Berger, Adam, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Boersma, Paul. 1998. *Functional Phonology.* Ph.D. thesis, University of Amsterdam.

Boersma, Paul and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry*, 32(1):45–86.

Boersma, Paul and Clara Levelt. 2000. Gradual constraint-ranking learning algorithm predicts acquisition order. In *Proceedings of Child Language Research Forum 30.* CSLI, Stanford, pages 229–237.

Della Pietra, Stephen, Vincent Della Pietra, and John Lafferty. 1995. Inducing features of random fields. CMU Technical Report CMU-CS-1995-144.

Goldwater, Sharon and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In Jennifer Spenader, Anders Eriksson, and Östen Dahl, editors, *Proceedings of the Stockholm Workshop on Variation within Optimality Theory.* Stockholm University, pages 111–120.

Keller, Frank and Ash Asudeh. 2002. Probabilistic learning algorithms and Optimality Theory. *Linguistic Inquiry*, 33(2):225–244.

Levelt, Claartje and Ruben van de Vijver. 1998. Syllable types in cross-linguistic and developmental grammars. manuscript. available from the Rutgers Optimality Archive.

Prince, Alan and Paul Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Technical Report TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.