Linear Optimality Theory as a Model of Gradience in Grammar

Frank Keller
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
phone: +44-131-650-4407, fax: +44-131-650-4587
keller@inf.ed.ac.uk

Abstract

This paper provides an overview of Linear Optimality Theory (LOT), a variant of Optimality Theory (OT) designed for the modeling of gradient acceptability judgment data. We summarize the empirical properties of gradient data that have been reported in the experimental literature, and use them to motivate the design of LOT. We discuss LOT's notions of constraint competition and optimality, as well as a new formulation of ranking argumentation, which makes it possible to apply standard parameter estimation techniques to LOT. Then the LOT model is compared to Standard OT, to Harmonic Grammar, and to recently proposed probabilistic versions of OT.

1 Introduction

This paper provides an overview of Linear Optimality Theory (LOT), a variant of optimality theory initially proposed by Keller (2000) to model gradient linguistic data. It is important to note that LOT is a framework designed to account for gradient judgment data; as has been argued elsewhere in this volume (Crocker and Keller, 2006), gradience in processing data and in corpus data has different properties from gradience in judgment data, and it is unlikely that the two types of gradience can be accounted for in a single, unified framework.

The remainder of the paper is structured as follows. In Section 2, we summarize the empirical properties of gradient judgments that motivate the design of LOT. Section 3 defines the components of an LOT grammar, and introduces the LOT notions of constraint competition and optimality. Based on this, ranking argumentation is defined, an algorithm for computing constraint ranks is introduced, and a measure of model fit in LOT is defined. Finally, Section 4 provides a comparison with other variants of OT, particularly with Standard OT and with Harmonic Grammar. This section also contains a survey of more recent developments, such as Probabilistic OT and variants of OT based on maximum entropy models.

Gisbert Fanselow, Caroline Féry, Ralph Vogel, and Matthias Schlesewsky, eds., *Gradience in Grammar: Generative Perspectives*, 270–287. Oxford: Oxford University Press, 2006.

2 Empirical properties of gradient judgments

Reviewing experimental data covering a range of syntactic phenomena in several languages, Sorace and Keller (2005) identify a number of sources of gradience in grammar. The two central experimental findings according to Sorace and Keller (2005) are that constraints are ranked and that constraint violations are cumulative. Constraint ranking means that some constraint violations are significantly more unacceptable than others. Cumulativity means the multiple constraint violations are significantly more unacceptable than single violations. These properties seem to be fundamental to the explanation of gradient linguistic judgments and therefore should form the basis of a model of gradience in grammar. Cumulativity also accounts for the ganging up of constraints: multiple violations of lower ranked constraints can be as unacceptable as a single violation of a higher ranked constraint. Experimental results reported by Keller (2000) show that a ganging up effect can be observed for constraints on word order, extraction, and gapping.

Sorace and Keller (2005) list a range of other properties of gradient data: context effects, crosslinguistic effects, and developmental optionality. They claim that these properties make it possible to classify linguistic constraints into soft and hard constraints. While this is an interesting claim, it seems to us more controversial than cumulativity and ranking, which seem to be more generally accepted properties of gradient data. As LOT only relies on cumulativity and ranking, we will not discuss the other properties here.

3 Linear Optimality Theory

Linear Optimality Theory as proposed by Keller (2000) is a model of gradience that makes predictions about the relative grammaticality of linguistic structures. It builds on core concepts from Optimality Theory, a framework that is attractive for this purpose as it is equipped with a notion of competition that makes it possible formalize the interaction of linguistic constraints. Furthermore, OT provides a notion of constraint ranking that makes it possible to account for the fact that constraints differ in strength, i.e., that some constraints are more important than others for the overall well-formedness of a given linguistic structure.

Although LOT borrows central concepts (such as constraint ranking and competition) from Optimality Theory, it differs in two crucial respects from existing OT-based accounts. Firstly, it relies on the assumption that constraint ranks are represented as sets of numeric weights, instead of as partial orders. Secondly, it assumes that the grammaticality of a given structure is proportional to the sum of the weights of the constraints it violates. This means that OT's notion of strict domination is replaced with an linear constraint combination scheme (hence the name Linear Optimality Theory).[1]

Only a limited number of components of the OT architecture are affected by the switch to LOT. The changes concern only *HEval,* the function that evaluates the harmony of a

[1] An anonymous reviewer points out that cumulativity could also be implemented using the mechanism of local constraint conjunction used in standard OT, which restricts cumulativity to particular local domains. Local conjunction has the advantage that the occurrence of cumulative effects is still under the control of the linguist: a local conjunction must be defined explicitly.

candidate, and *Rank*, the ranking component. LOT does not affect assumptions concerning the input and the generation function *Gen*, the two components of an OT grammar that determine which structures compete with each other. Also the constraint component *Con*, i.e., formal apparatus for representing constraints and candidates is unaffected. The LOT approach is neutral in these respects, and compatible with the diverse assumptions put forward in the OT literature.

However, LOT's versions of *HEval* and *Rank* entail changes in the way the optimal candidate is computed, as well as requiring a new type of ranking argumentation, i.e., a method for establishing constraint ranks from a set of linguistic examples. It will be shown that this type of ranking argumentation is considerably simpler than the one classically assumed in OT. Also, well understood algorithms exist for automating this type of ranking argumentation.

3.1 Violation profiles and harmony

The most prominent pattern in the experimental data presented by Keller (2000) is the *cumulativity* of constraint violations, i.e., the fact that the degree of unacceptability of a structure increases with the number of constraint violations it incurs. Cumulativity was in evidence in data on extraction, binding, gapping, and word order. Keller (2000) also shows that cumulativity effect extends from multiple violations of different constraints to multiple violations of the same constraint.

The other pervasive pattern in Keller's (2000) data is the *ranking* of constraints, i.e., the fact that constraint violations differ in the degree of unacceptability they cause. Constraint ranking was observed in data on extraction, binding, gapping, and word order.

The LOT model of gradient grammaticality derives from these two fundamental findings about constraint cumulativity and constraint ranking. Two hypotheses implement these two results. The first hypothesis deals with constraint ranking:

(1) **Ranking Hypothesis**
The ranking of linguistic constraints can be implemented by annotating each constraint with a numeric weight representing the reduction in acceptability caused by a violation of this constraint.

Note that this notion of constraint ranks as numeric weights is more general than the notion of ranks standardly assumed in Optimality Theory. Standard OT formulates constraint ranks as binary ordering statements of the form $C_1 \gg C_2$, meaning that constraint C_1 is ranked higher than the constraint C_2. Such statements do not make any assumptions regarding *how much* higher the ranking of C_1 is compared to the ranking of C_2. Such information is only available once we adopt a numeric concept of constraint ranking.

In the remainder of this paper, we will adopt the following terminological convention. The term constraint *weight* will be used to refer to the numeric annotation that our model assigns to a constraint. The term constraint *rank* will be employed to refer to the relative weight of two constraints in our model: we say that a constraint outranks another constraint if it has a greater weight (see also Definition (9) below). This usage is justified by the fact that Standard OT ranks (i.e., constraint orderings) are a special case of ranks as defined in Linear Optimality Theory (this will be shown in Section 4.1).

Once numeric constraint weights have been postulated, the overall acceptability of a structure can be computed based on the weights of the constraints that the structure violates. We will assume that simple summation is sufficient to compute the degree of acceptability of a structure from the weights of the constraints that the structure violates. This will account straightforwardly for the cumulativity of constraint violations observed experimentally. Keller (2000) demonstrates that this approach achieves a good model fit on his experimental data.

To account for the cumulativity of constraint weights, LOT formulates the Linearity Hypothesis in (2):

(2) **Linearity Hypothesis**
The cumulativity of constraint violations can be implemented by assuming that the grammaticality of a structure is proportional to the weighted sum of the constraint violations it incurs, where the weights correspond to constraint ranks.

The hypotheses in (1) and (2) can be made explicitly by formulating a numeric model that relates constraints ranks and degree of grammaticality. This relies on the notion of a grammar signature, which specifies the constraint set and the associated weights for a grammar. (Note that this definition, and all subsequent ones, are independent of the formulation of the constraints proper; the LOT account is one of constraint interaction, not of actual linguistic constraints.)

(3) **Grammar Signature**
A grammar signature is a tuple $\langle \mathbf{C}, w \rangle$ where $\mathbf{C} = \{C_1, C_2, \ldots, C_n\}$ is the constraint set, and $w(C_i)$ is a function that maps a constraint $C_i \in \mathbf{C}$ on its constraint weight w_i.

Relative to a grammar signature, a given candidate structure has a constraint violation profile as defined in (4). The violation specifies which constraints are violated by the structure and how often. This is a useful auxiliary notion that will be relied on in further definitions.

(4) **Violation Profile**
Given a constraint set $\mathbf{C} = \{C_1, C_2, \ldots, C_n\}$, the violation profile of a candidate structure S is the function $v(S, C_i)$ that maps S on the number of violations of the constraint $C_i \in \mathbf{C}$ incurred by S.

Based on Definitions (3) and (4), the harmony of a structure can now be defined using a simple linear model:

(5) **Harmony**
Let $\langle \mathbf{C}, w \rangle$ be a grammar signature. Then the harmony $H(S)$ of a candidate structure S with a violation profile $v(S, C_i)$ is given in (6).

(6) $$H(S) = -\sum_i w(C_i) v(S, C_i)$$

Equation (6) states that the harmony of a structure is the negation of the weighted sum of the constraint violations that the structure incurs. Intuitively, the harmony of a structure describes its degree of well-formedness relative to a given set of constraints. This notion corresponds closely to the definition of harmony assumed in Standard OT (Prince and Smolensky, 1997, p. 1607) or Harmonic Grammar (Smolensky et al., 1992, p. 14).

The assumption is that all constraint weights are positive, i.e., that $w_i \geq 0$ for all i. This means that only constraint violations influence the harmony of a structure. Constraint satisfactions will not change the harmony of the structure (including cases where a constraint is vacuously satisfied because it is not applicable). This assumption is in accordance with Keller's (2000) experimental results, in which only constraints violations were found to affect acceptability. This will discussed further in Section 4.2.

3.2 Constraint competition and optimality

Based on the definitions of violation profile and harmony proposed in the preceding section, LOT's notion of grammaticality can now be specified. Grammaticality is computed in terms of the relative harmony of two candidates in the same candidate set:

(7) **Grammaticality**
Let S_1 and S_2 be candidate structures in the candidate set **R**. Then S_1 is more grammatical than S_2 if $H(S_1) > H(S_2)$. This can be abbreviated as $S_1 > S_2$.[2]

A crucial difference between harmony and grammaticality follows from Definition (7). Harmony is an absolute notion that describes the overall well-formedness of a structure. Grammaticality, on the other hand, describes the relative ill-formedness of a structure compared with another structure. While it is possible to compare the harmony of two structures across candidates sets, the notion of grammaticality is only well-defined for two structures that belong to the same candidate set (i.e., share the same input). Therefore, Definition (7) (and the subsequent Definition (8)) provide a *relative* notion of well-formedness, in line with the optimality theoretic tradition.

Based on the definition of grammaticality in (7), we can define the optimal structure in a candidate set as the one with the highest relative grammaticality.

(8) **Optimality**
A structure S_{opt} is optimal in a candidate set **R** if $S_{opt} > S$ for every $S \in \mathbf{R}$.

A notion of constraint rank can readily be defined in LOT based on the relative weight of two constraints (see also the terminological note on ranks vs. weights in Section 3.1 above):

(9) **Constraint Rank**
A constraint C_1 outranks a constraint C_2 if $w(C_1) > w(C_2)$. This can be abbreviated as $C_1 \gg C_2$.

In what follows, we will illustrate the definitions for harmony, grammaticality, and optimality. Consider an example grammar with the constraints C_1, C_2, and C_3, and the constraints weights given in Table 1. This table also specifies an example candidate set $S_1, \ldots, S_4$ and gives the violation profiles for these candidates. The harmony for each of these structures can be computed based on Definition (5).

[2] This usage differs from the standard OT usage, where harmonic ordering is denoted by "$\succ$", not "$>$".

Table 1: Example violation profile and harmony scores

	C_1	C_2	C_3	
$w(C)$	4	3	1	$H(S)$
S_1		*	*	-4
S_2		*	**	-5
S_3			*	-1
S_4	*			-4

The structure S_3 maximizes harmony, i.e., it incurs the least serious violation profile. It is therefore the optimal structure in the candidate set, i.e., it is more grammatical than all other candidate structures. The structures S_1 and S_4 are both less grammatical than S_3. S_1 and S_4 receive the same harmony scores, but for different reasons; S_4 because it incurs a high-ranked violation of C_1, S_1 because it accumulates violations of C_2 and C_3. The structure S_2 is less grammatical than S_1, as it incurs an additional violations of C_3. In total, we obtain the following grammaticality hierarchy: $S_3 > \{S_1, S_4\} > S_2$.

This examples illustrates the three central properties of constraint interaction that were identified in Section 2. The first property is the *ranking* of constraints. S_3 incurs a violation of C_3, while S_4 incurs a violation of C_1. That S_3 is more grammatical than S_4 is accounted for by the fact that C_1 has a higher weight than C_3, i.e., the ranking $C_1 \gg C_3$ holds. This is a situation that was observed many times in the experimental data presented by Keller (2000).

Furthermore, the example illustrates how the *cumulativity* of constraint violations is modeled. S_1 incurs single violations of C_2 and C_3. The structure S_2 also incurs a single violation of C_2, but a double violation of C_3. As a consequence, S_1 is more grammatical than S_2. Cumulativity effects such as these encountered frequently in Keller's (2000) experimental data.

Finally, Table 1 illustrates the *ganging up* of constraint violations. The structures S_1 and S_4 have different constraint profiles: S_4 violates the constraint C_1, while S_1 violates the two constraints C_2 and C_3, which are both lower ranked than C_1. However, S_1 and S_4 are equally grammatical because the two constraints C_2 and C_3 gang up against C_1, leading to the same harmony score in both structures. Again, this empirical patters is in evidence in Keller's (2000) experimental data.

Note that standard optimality theoretic evaluation of the candidate set in Table 1 leads to a different harmonic ordering: $S_3 > S_1 > S_2 > S_4$. If we assume a naive extension of Standard OT, then this order corresponds to the grammaticality order of the candidates. The naive extension assumes the strict domination of constraints, and therefore fails to model ganging up effects. Under this approach, there is no possibility for a joint violation of C_2 and C_3 to be as serious as a single violation of C_1, due to the ranking $C_1 \gg C_2 \gg C_3$. Hence the naive extension of Standard OT fails to account for the ganging up effects that were observed experimentally.

3.3 Ranking argumentation and parameter estimation

Optimality Theory employs so-called *ranking arguments* to establish constraint rankings from data. A ranking argument refers to a set of candidate structures with a certain constraint violation profile, and derives a constraint ranking from this profile. This can be illustrated by the following example: assume that two structures S_1 and S_2 have the same constraint profile, with the following exception: S_1 violates constraint C_1, but satisfies C_2. Structure S_2, on the other hand, violates constraint C_2, but satisfies C_1. If S_1 is acceptable but S_2 is unacceptable, then we can conclude that the ranking $C_2 \gg C_1$ holds (see Prince and Smolensky 1993, 106).

In the general case, the fact that S_1 is acceptable but S_2 is unacceptable entails that each constraint violated by S_1 is outranked by at least one constraint violated by S_2. (See Hayes 1997, for a more extensive discussion of the inference patterns involved in ranking argumentation in Standard OT.)

The LOT approach allows a form of ranking argumentation that relies on gradient acceptability data instead of the binary acceptability judgments used in Standard OT. A ranking argument in Linear Optimality Theory can be constructed based on the difference in acceptability between two structures in the same candidate set, using the following definition:

(10) **Ranking Argument**
Let S_1 and S_2 be candidate structures in the candidate set $\mathbf{R}$ with the acceptability difference ΔH. Then the equation in (11) holds.

(11) $H(S_1) - H(S_2) = \Delta H$

This definition assumes that the difference in harmony between S_1 and S_2 is accounted for by ΔH, the acceptability difference between the two structures. ΔH can be observed empirically, and measured, for instance, using magnitude estimation judgments (Sorace and Keller, 2005). Drawing on the definition of harmony in (5), Equation (11) can be transformed to:

(12) $\sum_i w(C_i)(v(S_1,C_i) - v(S_2,C_i)) = -\Delta H$

This assumes that S_1 and S_2 have the violation profiles $v(S_1)$ and $v(S_2)$ and are evaluated relative to the grammar signature $\langle \mathbf{C}, w \rangle$.

Typically, a single ranking argument is not enough to rank the constraints of a given grammar. Rather, we need to accumulate a sufficiently large set of ranking arguments, based on which we can then deduce the constraint hierarchy of the grammar. To obtain a maximally informative set of ranking arguments, we take all the candidate structures in a given candidate set and compute a ranking argument for each pair of candidates, using Definition (12).

The number of ranking arguments that a set of k candidates yields is given in (13); note that this is simply the number of all unordered pairs that can be generated from a set of k elements.

(13) $n = \dfrac{k^2 - k}{2}$

Now we are faced with the task of computing the constraint weights of a grammar from a set of ranking arguments. This problem can be solved by regarding the set of ranking arguments as a system of linear equations. The solution for this system of equations will then provide a set of constraint weights for the grammar. This idea is best illustrated using an example. We consider the candidate set in Table 1 and determine all ranking arguments generated by this candidate set (here w_i is used as a shorthand for $w(C_i)$, the weight of constraint C_i):

(14)
$$\begin{array}{lrcrcr}
S_1 - S_2: & 0w_1 + 1w_2 + 1w_3 - 0w_1 - 1w_2 - 2w_3 & = & -((-4) - (-5)) & = & -1 \\
S_1 - S_3: & 0w_1 + 1w_2 + 1w_3 - 0w_1 - 0w_2 - 1w_3 & = & -((-4) - (-1)) & = & 3 \\
S_1 - S_4: & 0w_1 + 1w_2 + 1w_3 - 1w_1 - 0w_2 - 0w_3 & = & -((-4) - (-4)) & = & 0 \\
S_2 - S_3: & 0w_1 + 1w_2 + 2w_3 - 0w_1 - 0w_2 - 1w_3 & = & -((-5) - (-1)) & = & 4 \\
S_2 - S_4: & 0w_1 + 1w_2 + 2w_3 - 1w_1 - 0w_2 - 0w_3 & = & -((-5) - (-4)) & = & 1 \\
S_3 - S_4: & 0w_1 + 0w_2 + 1w_3 - 1w_1 - 0w_2 - 0w_3 & = & -((-1) - (-4)) & = & -3
\end{array}$$

This system of linear equations can be simplified to:

(15)
$$\begin{array}{lcr}
-w_3 & = & -1 \\
w_2 & = & 3 \\
w_2 + w_3 - w_1 & = & 0 \\
w_2 + w_3 & = & 4 \\
w_2 + 2w_3 - w_1 & = & 1 \\
w_3 - w_1 & = & -3
\end{array}$$

We have therefore determined that $w_2 = 3$ and $w_3 = 1$. The value of w_1 can be easily be obtained from any of the remaining equations: $w_1 = w_2 + w_3 = 4$.

This example demonstrates how a system of linear equations that follows from a set of ranking arguments can be solved by hand. However, such a manual approach is not practical for large systems of equations as they occur in realistic ranking argumentation. Typically, we will be faced with a large set of ranking arguments, generated by a candidate set with many structures, or by several candidate sets.

There are a number of standard algorithms for solving systems of linear equations, which can be utilized for automatically determining the constraint weights from a set of ranking arguments. One example is Gaussian Elimination, an algorithm which delivers an exact solution of a system of linear equations (if there is one). If we are dealing with experimental data, then the set of ranking arguments derived from a given data set will often result in an inconsistent set of linear equations, which means that Gaussian Elimination is not applicable. In such a case, the algorithm of choice is Least Square Estimation (LSE), a method for solving a system of linear equations even if the system is inconsistent. This means that LSE enables us to estimate the constraint weights of an LOT grammar if there is no set of weights that satisfy all the ranking arguments exactly (in contrast to Gaussian elimination). LSE will find an approximate set of constraint weights that maximizes the fit with the experimentally determined acceptability scores. A more detailed explanation of LSE and its application to LOT is provided by Keller (2000).

4 Comparison with other Optimality Theoretic approaches

4.1 Standard Optimality Theory

Linear Optimality Theory preserves key concepts of Standard Optimality Theory. This includes the fact that constraints are violable, even in an optimal structure. As in Standard OT, LOT avails itself of a notion of constraint ranking to resolve constraint conflicts; LOT's notion of ranking is quantified, i.e., richer than the one in Standard OT. The second core OT concept inherited by LOT is constraint competition. The optimality of a candidate cannot be determined in isolation, but only relative to other candidates it competes with. Furthermore, LOT uses ranking arguments in a similar way as Standard OT. Such ranking arguments work in a competitive fashion, i.e., based on the comparison of the relative grammaticality of two structure in the same candidate set. As in Standard OT, a comparison of structures across candidate sets is not well-defined; two structures only compete against each other if they share the same input.

The crucial difference between LOT and Standard OT is the fact that in LOT, constraint ranks are implemented as numeric weights and a straightforward linear constraint combination scheme is assumed. Standard Optimality Theory can then be regarded as a special case of LOT, where the constraint weights are chosen in an exponential fashion so as to achieve strict domination (see the Subset Theorem in (16)). The extension of Standard OT to LOT is crucial in accounting for the cumulativity of constraint violations. The linear constraint combination schema also greatly simplifies the task of determining a constraint hierarchy from a given data set. This problem simply reduces to solving a system of linear equations, a well-understood mathematical problem for which a set of standard algorithms exists (see Section 3.3).

Another advantage is that LOT naturally accounts for optionality, i.e., for cases where more than one candidate is optimal. Under the linearity hypothesis, this simply means that the two candidates have the same harmony score. Such a situation can arise if the two candidates have the same violation profile, or if they have different violation profiles, but the weighted sum of the violation is the same in both cases. No special mechanism for dealing with constraint ties are required in Linear OT. This is an advantage over Standard OT, where the modeling of optionality is less straightforward (see Asudeh 2001, for a discussion).

An OT grammar can be formulated as a weighted grammar if the constraint weights are chosen in an exponential fashion, so that strict domination of constraints is assured. This observation is due to Prince and Smolensky (1993, p. 200) and also applies to Linear Optimality Theory. Therefore, the theorem in (16) holds (the reader is referred to Keller (2000) for a proof).

(16) **Subset Theorem**
A Standard Optimality Theory grammar G with the constraint set $\mathbf{C} = \{C_1, C_2, \ldots, C_n\}$ and the ranking $C_n \gg C_{n-1} \gg \ldots \gg C_1$ can be expressed as a Linear Optimality Theory grammar G' with the signature $\langle \mathbf{C}, w \rangle$ and the weight function $w(C_i) = b^i$, where $b - 1$ is an upper bound for multiple constraint viola-

tions in G.

Note that the Subset Theorem holds only if there is an upper bound $b-1$ that limits the number of multiple constraint violations that the grammar G allows. Such an upper bound exists if we assume that the number of violations incurred by each structure generated by G is finite. This might not be true for all OT constraint systems.

4.2 Harmonic Grammar

Harmonic Grammar (Legendre et al., 1990a,b, 1991; Smolensky et al., 1992, 1993) is predecessor of OT that builds on the assumption that constraints are annotated with numeric weights (instead of just being rank-ordered as in Standard OT). Harmonic Grammar (HG) can be implemented in a hybrid connectionist-symbolic architecture and has been applied successfully to gradient data by Legendre et al. (1990a,b). As Prince and Smolensky (1993, p. 200) point out, "Optimality Theory [...] represents a very specialized kind of Harmonic Grammar, with exponential weighting of the constraints".

Linear Optimality Theory is similar to HG in that it assumes constraints that are annotated with numeric weights, and that the harmony of a structure is computed as the linear combination of the weights of the constraints it violates. There are, however, two differences between LOT and HG: (a) LOT only models constraint violations, while HG models both violations and satisfactions; and (b) LOT uses standard least square estimation to determine constraint weights, while HG requires more powerful training algorithms such as backpropagation. We will discuss each of these differences in turn.

LOT requires that all constraints weights have the same sign (only positive weights are allowed, see Section 3.1). This amounts to the claim that only constraint violations (but not constraint satisfactions) play a role in determining the grammaticality of a structure. In HG, in contrast, arbitrary constraint weights are possible, i.e., constraint satisfactions (as well as violations) can influence the harmony of a structure. This means that HG allows to define a grammar that contains a constraint C with the weight w and a constraint C' that is the negation of C and has the weight $-w$. In such a grammar, both the violations and the satisfactions of C influence the harmony of a structure.

The issue of positive weights has important repercussions for the relationship between Standard OT and LOT: Keller (2000) proves a Superset Theorem that states that an arbitrary LOT grammar can be simulated by a Standard OT grammar with stratified hierarchies. The proof crucially relies on the assumption that all constraint weights are of the same sign. Stratified hierarchies allow us to simulate the addition of constraint violations (they correspond to multiple violations in Standard OT), but they do not allow us to simulate the subtraction of constraint violations (which would be required by constraints that increase harmony). This means that the Superset Theorem does not hold for grammars that have both positive and negative constraints weights, as they are possible in Harmonic Grammar.

The second difference between HG and LOT concerns parameter estimation. An HG model can be implemented as a connectionist network, and the parameters of the model (the constraint weights) can be estimated using standard connectionist training algorithms. An example is provided by the HG model of unaccusativity/unergativity in French presented by Legendre et al. (1990a,b) and Smolensky et al. (1992). This model is implemented as a multilayer perceptron and trained using the backpropagation algorithm (Rumelhart et al., 1986).

It is well known that many connectionist models have an equivalent in conventional statistical techniques for function approximation. Multilayer perceptrons, for instance, correspond to a family of non-linear statistical models, as shown by Sarle (1994). (Which non-linear model a given perceptron corresponds to depends on its architecture, in particular the number and size of the hidden layers.) The parameters of a multilayer perceptron are typically estimated using backpropagation or similar training algorithms.

On the other hand, a single-layer perceptron (i.e., a perceptron without hidden layers) corresponds to multiple linear regression, a standard statistical technique for approximating a linear function of multiple variables. The parameters (of both a single-layer perceptron and a linear repression model) can be computed using least square estimation (Bishop, 1995). This technique can also be used for parameter estimation for LOT models (see Section 3.3). Note that LOT can be conceived of as a variant of multiple linear regression. The difference between LOT and conventional multiple linear regression is that parameter estimation is not carried directly on data to be accounted for (the acceptability judgments); rather, a preprocessing step is carried out on the judgment data to compute a set of ranking arguments, which then form the input for the regression.

To summarize, the crucial difference between HG and LOT is that HG is a non-linear function approximator, while LOT is a linear function approximator, i.e., a variant of linear regression. This means that a different set of parameter estimation algorithms is appropriate for HG and LOT, respectively.

4.3 Probabilistic Optimality Theory

Boersma and Hayes (2001) propose a probabilistic variant of Optimality Theory (POT) that is designed to account for gradience both in corpus frequencies and in acceptability judgments. POT stipulates a continuous scale of *constraint strictness*. Constraints are annotated with numerical strictness values; if a constraint C_1 has a higher strictness value that a constraint C_2, then C_1 outranks C_2. Boersma and Hayes (2001) assume *probabilistic constraint evaluation*, which means that at evaluation time, a small amount of random noise is added to the strictness value of a constraint. As a consequence, *re-rankings* of constraints are possible if the amount of noise added to the strictness values exceeds the distance between the constraints on the strictness scale.

For instance, assume that two constraints C_1 and C_2 are ranked $C_1 \gg C_2$, selecting the structure S_1 as optimal for a given input. Under Boersma and Hayes's (2001) approach, a re-ranking of C_1 and C_2 can occur at evaluation time, resulting in the opposite ranking $C_2 \gg C_1$. This re-ranking might result in an alternative optimal candidate S_2. The probability of the re-ranking that makes S_2 optimal depends on the distance between C_1 and C_2 on the strictness scale (and on the amount of noise added to the strictness values). The re-ranking probability is assumed to predict the degree of grammaticality of S_2. The more probable the re-ranking $C_2 \gg C_1$, the higher the degree of grammaticality of S_2; if the rankings $C_1 \gg C_2$ and $C_2 \gg C_1$ are equally probable, then S_1 and S_2 are equally grammatical.

Table 2: Data that cannot be modeled in Probabilistic OT (hypothetical frequencies or acceptability scores), from Keller and Asudeh (2002)

/input/	C_3	C_1	C_2	Freq./Accept.
S_1		*		3
S_2		*	*	2
S_3	*			1

Table 3: Data that cannot be modeled in Probabilistic OT (hypothetical frequencies or acceptability scores), from Keller and Asudeh (2002)

/input/	C_1	C_2	Freq./Accept.
S_1		*	4
S_2		**	3
S_3		***	2
S_4	*		1

The POT framework comes with its own learning theory in the form of the Gradual Learning Algorithm (Boersma, 1998, 2000; Boersma and Hayes, 2001). This algorithm is a generalization of Tesar and Smolensky's (1998) Constraint Demotion Algorithm in that it performs constraint promotion as well as demotion. The Gradual Learning Algorithm incrementally adjusts the strictness values of the constraints in the grammar to match the frequencies of the candidate structures in the training data. The fact that the algorithm relies on gradual changes makes it robust to noise, which is an attractive property from a language acquisition point of view.

There are, however, a number of problems with the POT approach. As Keller and Asudeh (2002) point out, POT cannot model cases of harmonic bounding, as a illustrated in Table 2: candidate S_2 is harmonically bound by candidate S_1, which means that there is no re-ranking of the constraints that would make S_2 optimal. As S_2 can never be optimal, its frequency or acceptability is predicated to be zero (i.e., no other candidate can be worse, even if it violates additional constraints). An example where this is clearly incorrect is S_3 in Table 2, which violates a higher ranked constraint and is less acceptable (or less frequent) than S_2.

A second problem with POT identified by Keller and Asudeh (2002) is cumulativity. This can be illustrated with respect to Table 3: here, candidate S_1 violates constraint C_2 once and is more acceptable than S_2, which violates C_2 twice. S_2 in turn is more acceptable than S_3, which violates C_2 three times. A model based on constraint re-ranking cannot account for this, as a re-ranking of C_2 will not change the outcome of the competition between S_1, S_2, and S_3. Essentially, this is a special case of harmonic bounding involving only one constraint.

There is considerable evidence that configurations such as the ones illustrated in Tables 2 and 3 occur in real data. Keller (2000) reports acceptability judgment data for word

Table 4: Data that cannot be modeled in POT′ (hypothetical frequencies or acceptability scores)

/input/	C_3	C_1	C_2	Freq./Accept.
S_1		*		2
S_2		*	*	1
S_3	*			1

order variation in German that instantiates both patterns. Guy and Boberg's (1997) frequency data for coronal stop deletion in English instantiates the cumulative pattern in Table 3. Jäger and Rosenbach (2004) show that cumulativity is instantiated in both frequency data and acceptability data on genitive formation in English. None of these data sets can be modeled by POT, and thus constitute serious counterexamples to this approach. In Linear Optimality Theory, on the other hand, such cases are completely unproblematic, due to the linear combination scheme assumed in this framework.

In a recent paper, Boersma (2006) acknowledges that cases of harmonic bounding and cumulativity as illustrated in Tables 2 and 3 pose a problem for POT. In response to this, he proposes a variant of POT, which we will call POT′. In POT′, the acceptability of a candidate S is determined by carrying out a pairwise comparison between S and each of the other candidates in the candidate set; the acceptability of S then corresponds to the percentage of comparisons that S wins.[3] As an example, take the tableau in Table 2. Here, S_1 wins against S_2 and S_3, hence his acceptability value is $2/2 = 100\%$. S_2 wins against S_3 but loses against S_1, so its acceptability is $1/2 = 50\%$. S_3 loses against both candidates, and thus receives an acceptability value of 0%.

In POT′, the relative grammaticality of a candidate corresponds to its optimality theoretic rank in the candidate set. This is not a new idea; in fact it is equivalent to the definition of relative grammaticality in terms of *suboptimality*, initially proposed by Keller (1997). The only difference is that in POT′, suboptimality is determine based on a POT notion of harmony, instead of using the standard OT notion of harmony, as assumed by Keller (1997). However, there are a number of conceptual problems with this proposal (which carry over to POT′), discussed in detail by Müller (1999) and Keller (2000).

In addition to that, there are empirical problems with the POT′ approach. POT′ correctly predicts the relative acceptability of the example in Table 2 (as outlined above). However, other counterexamples can be constructed easily if we assume *ganging up effects*. In Table 4, the combined violation of C_1 and C_2 is as serious as the single violation of C_3, which means that the candidates S_2 and S_3 are equally grammatical. Such a situation can not be modeled in POT′, as S_2 will win against S_3 (because C_3 outranks C_1), hence S_2 is predicted to be more grammatical than S_3. As discussed in Section 2, ganging up effects occur in experimental data, and thus pose a real problem for POT′.

In contrast to POT and POT′, LOT can model ganging up effects straightforwardly, as illustrated in Section 3.2. This is not surprising: the weights in LOT grammars are

[3] More precisely, it is the POT probability of winning, averaged over all pairwise comparisons, but this difference is irrelevant here.

estimated so that they correspond in a linear fashion to the acceptability scores of the candidates in the training data. The strictness bands in POT (and POT′) grammars, on the other hand, are estimated to match the frequencies of candidates in the training data; it is not obvious why such a model should correctly predict acceptability scores, given that it is trained on a different type of data.

4.4 Maximum entropy models

The problems with POT have led a number of authors to propose alternative ways of dealing with gradience in OT. Goldwater and Johnson (2003), Jäger (2004), and Jäger and Rosenbach (2004) propose a probabilistic variant of OT based on the machine learning framework of *maximum entropy models*, which is state of the art in computational linguistics (e.g., Abney 1997; Berger et al. 1996). In Maximum Entropy OT (MOT) as formulated by Jäger (2004), the probability of a candidate structure (i.e., of an input-output pair (o,i)) is defined as:

(17) $$p_R(o|i) = \frac{1}{Z_R(i)} \exp(\sum_j r_j c_j(i,o))$$

Here, r_j denotes the numeric rank of constraint j, while R denotes the ranking vector, i.e., the set of ranks of all constraints. The function $c_j(i,o)$ returns the number of violations of constraint j incurred by input-output pair (i,o). $Z_R(i)$ is a normalization factor.

The model defined in (17) can be regarded as an extension of LOT as introduced in Section 3.1. It is standard practice in the literature on gradient grammaticality to model not raw acceptability scores, but log-transformed, normalized acceptability data (Keller, 2000). This can be made explicit by log-transforming the lefthand side of (6) (and dropping the minus and renaming the variable i to j). The resulting formula is then equivalent to (18).

(18) $$H(S) = \exp(\sum_j w(C_j) v(S,C_j))$$

A comparison of (17) and (18) shows that the two models have a parallel structure: $w(C_j) = r_j$ and $v(S,C_j) = c_j(i,o)$ (the input-output structure of the candidates is implicit in (18)). Both models are instances of a more general family of models referred to as log-linear models. There is, however, a crucial difference between the MOT definition in (17) and the LOT definition in (18). Equation (18) does not include the normalization factor $Z_R(i)$, which means that (18) does not express a valid probability distribution. The normalization factor is not trivial to compute, as it involves summing over all possible output forms o (see Goldwater and Johnson 2003, and Jäger 2004, for details). This is the reason why LOT can assumes a simple learning algorithm based on least square estimation, while MOT has to rely on learning algorithms for maximum entropy models, such as generalized iterative scaling, or improved iterative scaling (Berger et al., 1996). Another crucial difference between MOT and LOT (pointed out by Goldwater and Johnson 2003) is that MOT is designed to be trained on corpus data, while LOT is designed to be trained on acceptability judgment data.

5 Conclusions

This paper introduced Linear Optimality Theory (LOT) as model of gradient grammaticality. Although this model borrows central concepts (such as constraint ranking and competition) from Optimality Theory, it differs in two crucial respects from Standard OT. Firstly, LOT assumes that constraint ranks are represented as numeric weights (this feature is shared with Probabilistic OT and Maximum Entropy OT, see Sections 4.3 and 4.4). Secondly, LOT assumes that the grammaticality of a given structure is proportional to the sum of the weights of the constraints it violates, which means that OT's notion of strict domination is replaced with an linear constraint combination scheme (this feature is shared with Maximum Entropy OT, see Section 4.4).

We also outlined a learning algorithm for LOT (see Section 3.3). This algorithm takes as its input a grammar (i.e., a set of linguistic constraints) and a training set, based on which it estimates the weights of the constraints in the grammar. The training set is a collection of candidate structures, with the violation profile and the grammaticality score for each structure specified. Note that LOT is not intended as a model of human language acquisition: it cannot be assumed that the learner has access to training data that are annotated with acceptability scores. The sole purpose of the LOT learning algorithm is to perform parameter fitting for LOT grammars, i.e., to determine an optimal set of constraint weights for a given data set.

LOT is able to account for the properties of gradient structures discussed in Section 2. Constraint ranking is modeled by the fact that LOT annotates constraints with numeric weights representing the contribution of a constraint to the unacceptability of a structure. Cumulativity is modeled by the assumption that the degree of ungrammaticality of a structure is computed as the sum of the weights of the constraint the structure violates. Once ranking and cumulativity are assumed as part of the LOT model, other properties of gradient linguistic judgments follow without further stipulations.

References

Abney, Steven. 1997. Stochastic attribute-value grammars. *Computational Linguistics* 23(4): 597–618.

Asudeh, Ash. 2001. Linking, optionality, and ambiguity in Marathi. In Peter Sells, ed., *Formal and Empirical Issues in Optimality-theoretic Syntax*. Stanford, CA: CSLI Publications.

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

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.

Boersma, Paul. 1998. *Functional Phonology: Formalizing the Interactions between Articulatory and Perceptual Drives*. The Hague: Holland Academic Graphics.

—. 2000. Learning a grammar in functional phonology. In Joost Dekkers, Frank van der Leeuw, and Jeroen van de Weijer, eds., *Optimality Theory: Phonology, Syntax, and Acquisition*, 465–523. Oxford: Oxford Univeristy Press.

—. 2006. A Stochastic OT account of paralinguistic task such as grammaticality and prototypicality judgments. In Fanselow et al. 2006.

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

COGSCI. 1990. *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Lawrence Erlbaum Associates.

Crocker, Matthew W., and Frank Keller. 2006. Probabilistic grammars as models of gradience in language processing. In Fanselow et al. 2006, 227–245.

Fanselow, Gisbert, Caroline Féry, Ralph Vogel, and Matthias Schlesewsky, eds. 2006. *Gradience in Grammar: Generative Perspectives*. Oxford: Oxford University Press.

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

Guy, Gregory R., and Charles Boberg. 1997. Inherent variability and the obligatory contour principle. *Language Variation and Change* 9: 149–164.

Hayes, Bruce P. 1997. Four rules of inference for ranking argumentation. Unpubl. ms., Department of Linguistics, University of California, Los Angeles.

Jäger, Gerhard. 2004. Maximum entropy models and Stochastic Optimality Theory. Unpubl. ms., University of Potsdam.

Jäger, Gerhard, and Anette Rosenbach. 2004. The winner takes it all – almost: Cumulativity in grammatical variation. Unpubl. ms., University of Potsdam and University of Düsseldorf.

Keller, Frank. 1997. Extraction, gradedness, and optimality. In Alexis Dimitriadis, Laura Siegel, Clarissa Surek-Clark, and Alexander Williams, eds., *Proceedings of the 21st Annual Penn Linguistics Colloquium*, no. 4.2 in Penn Working Papers in Linguistics, 169–186. Department of Linguistics, University of Pennsylvania.

—. 2000. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. Ph.D. thesis, University of Edinburgh.

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

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990a. Harmonic grammar: A formal multi-level connectionist theory of linguistic well-formedness: An application. In COGSCI 1990, 884–891.

—. 1990b. Harmonic grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. In COGSCI 1990, 388–395.

—. 1991. Unifying syntactic and semantic approaches to unaccusativity: A connectionist approach. In *Proceedings of the 17th Annual Meeting of the Berkeley Linguistics Society*, 388–395. Berkeley.

Müller, Gereon. 1999. Optimality, markedness, and word order in German. *Linguistics* 37(5): 777–818.

Prince, Alan, and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Technical Report 2, Center for Cognitive Science, Rutgers University.

—. 1997. Optimality: From neural networks to universal grammar. *Science* 275: 1604–1610.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: Foundations, 318–362. Cambridge, MA: MIT Press.

Sarle, Warren S. 1994. Neural networks and statistical models. In *Proceedings of the 19th Annual SAS Users Group International Conference*, 1538–1550. SAS Institute, Cary, NC.

Smolensky, Paul, Géraldine Legendre, and Yoshiro Miyata. 1992. Principles for an integrated connectionist/symbolic theory of higher cognition. Report CU-CS-600-92, Computer Science Department, University of Colorado at Boulder.

—. 1993. Integrating connectionist and symbolic computation for the theory of language. *Current Science* 64: 381–391.

Sorace, Antonella, and Frank Keller. 2005. Gradience in linguistic data. *Lingua* 115(11): 1497–1524.

Tesar, Bruce, and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29(2): 229–268.