Learning with Violable Constraints

Gaja Jarosz
Yale University

0. Abstract

This chapter reviews major results as well as areas of ongoing research on computational models of learning with violable constraints, in the sense of Optimality Theory. The review encompasses learning in classic OT as well as learning in related frameworks that formalize constraint interaction as (probabilistic) weighting or ranking. The learning problem is decomposed into several subproblems, which are considered in turn. The discussion emphasizes the challenges posed by these various subproblems and the insights and differences of the proposed solutions. The chapter first discusses the narrow grammar-learning subproblem, the problem of learning rankings or weightings given data along with their full structural descriptions. Next, the discussion turns to several distinct approaches to the broader grammar-learning subproblem, which makes the more realistic assumption that hidden structure, such as prosodic structure or underlying representations, are not available to the learner and must themselves be learned. The chapter then discusses the subset problem and various approaches that have been proposed for learning restrictive constraint grammars. Finally, the chapter briefly reviews work on acquisition modeling, work that connects the predictions of computational models and the process by which children acquire language.

1. Introduction

Learnability is a central problem of linguistics. A complete theory of language must explain how this rich and complex system of knowledge can be learned from the limited available input. Work on learnability seeks to answer this question by developing explicit formal models of the learning process, models that provide concrete hypotheses about how language can be learned. Work on learnability within Optimality Theory (OT; Prince and Smolensky 1993 / 2004)¹ and related frameworks has contributed a great deal to our understanding of language learning. These results contribute not only to our understanding of how language might be learned in principle, but also to our understanding of how language is actually acquired by children. This chapter reviews these important contributions, focusing primarily on major learnability results and challenges, but also reviewing work that relates the predictions of computational models to developmental findings.

This chapter discusses the various facets of the learning problem, the challenges they pose for the learner, and the constraint-based approaches that have been developed to address them. Under the standard view in OT, the set of universal constraints defines, via permutation, the space of possible adult languages (the typology). This formal system defines the hypothesis space for learning – the set of languages that learning models take as the targets of learning. Under this view, the major subproblem of language learning is the learning of grammars (rankings or weightings of constraints) for a given set of constraints.

¹ The focus of this chapter is on learning, but there is also a lot of work on the important, related computational problems of generation and recognition. Learning and generation are intrinsically linked since most (all?) constraint-based learning models assume a generation mechanism. These problems are conceptually distinct, however, and the focus of the present paper is on learning (for related work on the generation and recognition problems, see e.g. Ellison 1994; Tesar 1995; Frank and Satta 1998; Eisner 2000; Eisner 2002; Riggle 2004; Idsardi 2006; Heinz, Kobele, and Riggle 2009).
The grammar-learning subproblem itself consists of a number of nontrivial subproblems. In the general case, the grammar must be learned from a set of unstructured overt forms to which the learner is exposed. Tesar and Smolensky (1998, 2000) decompose this larger grammar-learning problem into a number of processes, including the learning of grammars from full structural descriptions. Full structural descriptions are the representations evaluated by constraints in an OT grammar and include underlying/lexical representations as well as hidden structural information, such as prosodic or syntactic structure. Although this narrower grammar-learning subproblem is itself nontrivial, there now exist a number of solutions both for classic OT (Section 2) as well as for probabilistic and weighed extensions of OT (Section 3). Learnability results relating to the broader grammar-learning subproblem, which makes the more realistic assumption that learners do not have access to hidden structure, are discussed in Section 4. Section 5 briefly reviews results on an important aspect of language learning, the learning of restrictive languages. Finally, most learnability work takes into account basic considerations of psychological plausibility, such as the computational resources expected of the learner. Some computational work goes further in attempting to model aspects of the acquisition process itself and relating the behavior of the model to developmental findings - Section 6 reviews constraint-based modeling work of this sort.

Although most work on constraint-based computational modeling has focused within the domain of phonology, most of the models for constraint-based learning are entirely general, applying equally well to other linguistic domains. In other words, most of the learnability results depend only on the formal structure of the framework and not at all on the content of the constraints or the representations they evaluate. This contrasts with approaches to learning within parametric frameworks that rely crucially on identifying domain-specific cues for setting of individual parameters (Dresher and Kaye 1990; Dresher 1999). Another approach to learning in parametric frameworks relies on the existence of triggers, special kinds of data that can uniquely specify individual parameter settings (Gibson and Wexler 1994; Frank and Kapur 1996; Fodor 1998). As discussed by Tesar and Smolensky (1998, 2000), guaranteeing the existence of such triggers often requires restrictions on the grammar in order to ensure parameters are independent, a goal that unfortunately often conflicts with the goals of typological explanation. For extensive discussion comparing constraint-based and parametric approaches to the broader grammar-learning subproblem, see Tesar 2004b. Although constraint-based learning models are largely domain-general, they have been applied extensively to the learning of phonological grammars, especially the learning of phonological alternations. The discussion below will therefore focus on phonology; the extension of these learning models to syntax and other domains remains a promising direction for further research.

2. Learning Rankings

This section reviews proposals within Classic OT for the learning of a ranking of constraints consistent with a set of (fully structured) data. Tesar (1995) and Tesar and Smolensky (1998) developed a family of learning algorithms for the (narrow) grammar-learning subproblem within classic OT almost simultaneously with the introduction of OT. Section 2.1 discusses Recursive Constraint Demotion (RCD), an algorithm that learns rankings given a set of winner-loser pairs. Algorithms for extracting and processing these winner-loser pairs during online learning are discussed in Section 2.2.

2.1 Recursive Constraint Demotion

Learning a ranking entails finding a ranking of the constraints under which each learning datum is rendered optimal. In order for the datum to be optimal, every constraint that favors
some other candidate must be dominated by a constraint that favors the learner’s datum. The requirements behind this ranking logic are particularly transparent in Comparative Tableaux (Prince 2002a). Rather than showing constraint violations of individual candidates, each row of a Comparative Tableaux represents a winner-loser pair, a comparison between the constraint violations of the designated winner and another candidate, the loser. The winner-loser pair comparison indicates for each constraint whether it favors the winner (W), the loser (L), or has no preference (e). For example, if the winner is [pat], the constraint NoCODA assigns more violations to the winner [pat] than to the loser [pa], so this winner-loser pair would get an L for NoCODA. The L indicates that NoCODA must be dominated by some constraint with a W for this pair in order for [pat] to be optimal. Stated in terms of Comparative Tableaux, a ranking selects the winner as optimal if for every winner-loser pair, every loser-prefering constraint is dominated by a winner-prefering constraint. Put simply, each L must be preceded by a W in the same row.

Viewed in this way, RCD (Tesar 1995; Tesar and Smolensky 1998) is an efficient way of reordering the constraints such that each loser-prefering constraint is dominated by some winner-prefering constraint. RCD takes full advantage of strict domination by creating the ranking top down – once a candidate is ruled out by high-ranking constraints, it is no longer considered by the algorithm. It works recursively, repeatedly selecting constraints to rank in the hierarchy, removing them from the working set, and repeating the process. Each time a set of constraints is removed from the working set, it is placed in successively lower strata of the hierarchy. A constraint is available to rank in the hierarchy if it prefers no losers that are still in play in the working set (has no Ls in its column). Once removed and placed at the top of the hierarchy, constraints that are winner-prefering make new constraints available to rank by eliminating losing candidates from consideration, and the process repeats until no constraints are left to rank.

(1) Constraints
a. *+VOI/OBS – No voiced obstruents. One violation for every voiced obstruent.
b. ID[VOI] – No changes in voicing. One violation for every voicing feature changed from input to output.
c. AGREE[VOI] – Sequences of obstruents must agree in voicing specification. One violation for every pair of adjacent obstruents differing in voicing.
d. MAX – No deletion. One violation for every deleted segment.

(2) A set of winner-loser pairs before RCD

<table>
<thead>
<tr>
<th></th>
<th>*+VOI/OBS</th>
<th>ID[VOI]</th>
<th>AGREE[VOI]</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lupz/</td>
<td>[lups] ~ [lup]</td>
<td>e</td>
<td>L</td>
<td>e</td>
</tr>
<tr>
<td>/lupz/</td>
<td>[lups] ~ [lupz]</td>
<td>W</td>
<td>L</td>
<td>W</td>
</tr>
<tr>
<td>/dan/</td>
<td>[dan] ~ [tan]</td>
<td>L</td>
<td>W</td>
<td>e</td>
</tr>
</tbody>
</table>

The process is illustrated for the winner-loser pairs in (2), which rely on the constraints in (1). The first pass of RCD examines each of the four constraints, identifying AGREE[VOI] and MAX as the two constraints that prefer no losers. These two constraints are thus placed at the top of the hierarchy and removed. As shown in (3), these two constraints favor the winners of the first two winner-loser pairs, with AGREE[VOI] ruling out the losing candidate [lupz], and MAX ruling out the losing candidate [lup]. The high ranking of AGREE and MAX has ruled out these candidates so RCD removes them from consideration.
Understanding the result of the first pass of RCD

Thus, after the first pass of RCD there are just two constraints and one winner-loser pair left, as shown in (4). Now, RCD is called recursively on this resulting set of winner-loser pairs and reduced set of constraints and determines that ID[VOI] is the only constraint that prefers no losers, and so it is placed in the next highest stratum and removed. At this point the ranking that has been built is \{AGREE[VOI], MAX\} » ID[VOI].

Since ID[VOI] favors the winner [dan] for the remaining winner-loser pair, the pair is removed, leaving an empty set of winner-loser pairs. On the final pass of RCD, there are no losers left so all the remaining constraints, just *+VOI/OBS in this case, are placed in the ranking. The hierarchy found by RCD is thus \{AGREE[VOI], MAX\} » ID[VOI] » *+VOI/OBS, and any total ranking consistent with it is guaranteed to be consistent with all the winner-loser pairs.

In this example, which has four constraints, it takes RCD three passes to find a consistent ranking. In the general case, it will take at most $n$ passes of RCD to find a consistent ranking (assuming there is one) for $n$ constraints (Tesar 1995; Tesar and Smolensky 1998). This is because, if there is a consistent ranking, then each pass of RCD ranks at least one constraint and thus is complete within $n$ passes. How long each pass takes also depends on the number of constraints, but each pass cannot consider more than $n$ constraints. The whole procedure therefore cannot exceed $n^2$ steps, which compares very favorably with the size of the search space, e.g. the number of rankings, which is $n!$.

Another important feature of RCD is its ability to determine when no ranking is consistent with a set of winner-loser pairs. Inconsistency is detected if at any point during learning there remain unranked constraints, but all of them prefer some losers. Inconsistency detection is a crucial component of later proposals dealing with structural ambiguity (Tesar 1997b; Tesar 2004b) and the learning of underlying forms (Tesar, Alderete, Horwood, Merchant, Nishitani, and Prince 2003; Tesar 2006a; Tesar 2006b; Merchant 2008; Merchant and Tesar 2008; Tesar 2009), which are discussed in Section 4.

In addition to RCD, Tesar (1995) and Tesar and Smolensky (1998) discuss a related learning algorithm, Constraint Demotion (CD), which is the basis for the well-known Error-Driven Constraint Demotion, discussed in the next section. CD examines one winner-loser pair at a time and places all loser-prefering constraints ranked above the highest-ranked winner-prefering constraint to a stratum immediately below the highest-ranked winner-prefering constraint. In other words, it demotes high-ranking loser-prefering constraints so that all Ls end up below a W. The computational properties of CD are similar to RCD: CD is also guaranteed to efficiently find a correct ranking if there is one (see Tesar 1995 for in depth comparisons). The main advantage of CD is that it can be applied to one winner-loser pair at a time. In contrast to RCD, however, CD cannot detect inconsistency. The next section
discusses variants of both of these algorithms, which process one learning datum at a time, relying on error-driven learning.²

2.2 Error-Driven Learning

Given a consistent set of winner-losers pairs, RCD is guaranteed to efficiently find a correct ranking; however, the procedure itself does not specify where these winner-loser pairs come from. Given full structural descriptions for the learning data, winners are simply the observed forms in the learning data. Error-driven learning (Wexler and Culicover 1980; Tesar 1995; Tesar and Smolensky 1998) provides a particularly efficient way of selecting informative losers. In error-driven learning, the learner processes each learning datum to determine what candidate is considered optimal under the current ranking. Specifically, the learner uses the underlying form provided in the full structural description of the datum and its current ranking to generate its output for that datum. If the learner’s output matches the learning datum, the current ranking already correctly accounts for that learning datum, and the learner does nothing. If, however, the learner’s output does not match the learning datum, it is considered an error and triggers re-ranking. The learner creates a winner-loser pair using the learning datum and the learner’s own (incorrect) output, and this pair is used to update the ranking hypothesis. In this way, the learner only considers winner-loser pairs that are informative because they indicate that a change to the current ranking is necessary: the (potentially infinite) space of possible candidates is never directly consulted during learning. Learning continues until no more errors are produced. This section describes two error-driven learners for the Classic OT (narrow) grammar-learning subproblem, Error-Driven Constraint Demotion (Tesar 1995; Tesar and Smolensky 1998) and Multi-Recursive Constraint Demotion (Tesar 1997b; Tesar 2004b). Other error-driven learners for Classic OT are discussed by Boersma (1998) and Magri (2009).

Error-Driven Constraint Demotion (EDCD) maintains a single grammar hypothesis, processes one learning datum at a time, and if an error is produced, applies CD to the resulting winner-loser pair. EDCD is an online learning algorithm since it processes one datum at a time. EDCD also has efficient data complexity: it needs to process at most \( \frac{1}{2}n(n-1) \) informative winner-loser pairs before settling on a final ranking (Tesar 1995; Tesar and Smolensky 1998). Finally, just like CD, EDCD is guaranteed to find a ranking consistent with the set of winner-loser pairs it considers, if there is one³ (Tesar 1995; Tesar and Smolensky 1998).

The appeal of EDCD is its correctness, efficiency, and simplicity; however, because EDCD cannot detect inconsistency, Tesar later proposed an error-driven variant of RCD called Multi-Recursive Constraint Demotion (Tesar 1997b; Tesar 2004b). Multi-Recursive Constraint Demotion (MRCD) is also online since it processes one learning datum at a time. Unlike EDCD, however, MRCD keeps track of all the winner-loser pairs it encounters. Specifically, rather than maintaining a single ranking and using it to process each learning datum, MRCD maintains a list of winner-loser pairs, called a support, that it uses to process each new datum. For each learning datum, MRCD applies RCD to the support in order to construct a hierarchy, uses the hierarchy to generate the learner’s output for the datum, and if

² For alternative ways of constructing losing competitors, see Riggle 2004.
³ EDCD is guaranteed to find a ranking consistent with all the winner-loser pairs it encounters; however, as discussed by Tesar (2000a) and Merchant and Tesar (2008), EDCD can fail to find all the necessary winner-loser pairs as a result of the treatment of ties. Thankfully, a simple adjustment to EDCD’s original treatment of tied constraints fixes this issue. Tesar (2000a) proposes to replace the ‘pooling ties’ of the original EDCD with ‘conflict ties’, which generate an error if there is crucial conflict between constraints that are tied. Boersma (2009) proposes an alternative solution, ‘permuting ties’. Finally, Merchant and Tesar (2008) suggest another possibility, relying on Riggle’s (2004) Contenders algorithm to generate losers.
Learning with Violable Constraints – Gaja Jarosz

an error is produced, adds the resulting winner-loser pair to the support. Just like EDCD, MRCD has efficient data complexity, and never needs to consider more than \( \frac{1}{2}n(n-1) \) winner-loser pairs before settling on a ranking that produces no more errors (Tesar 1997b; Tesar 2004b). Storing the list of winner-loser pairs does increase the amount of computational effort expected of the learner, and it means the learner applies the full RCD algorithm during processing of each datum, but it also enables the learner to detect when a new winner-loser pair is inconsistent with the existing list. The relevance of this feature of MRCD will become clear in the context of learning with hidden structure, discussed in Section 4.

In sum, nearly simultaneously with the introduction of OT, Tesar (1995) and Tesar and Smolensky (1998) introduced a family of learning algorithms for the narrow grammar-learning subproblem of Classic OT. The Constraint Demotion family includes RCD, which is guaranteed to efficiently find a ranking consistent with a set of winner-loser pairs or to detect inconsistency, if there is no consistent ranking. The online, error-driven variant of RCD is called MRCD. The family of algorithms also includes CD, which processes one winner-loser pair at a time, and whose online, error-driven variant is called EDCD. These learning algorithms form a foundation for subsequent work on larger subproblems for Classic OT as well as for a number of error-driven learning algorithms for probabilistic extensions of OT. Approaches to the narrow grammar-learning subproblem relying on probabilistic and weighted generalizations of OT are discussed next.

3. Beyond Total Ranking

This section discusses two orthogonal ways of generalizing classic OT and the application of these generalizations to the grammar-learning subproblem. One way that the classic OT ranking can be generalized is by assuming that grammars are actually mixtures of multiple total rankings, or probability distributions over rankings. The notion that an individual’s grammar may consist of multiple rankings has played a prominent role in the study of variation and optionality (see Coetzee and Pater 2008b and Anttila 2007 for overviews). Under this view, variation arises when an individual’s grammar varies stochastically between multiple total rankings, with different rankings selecting different candidates as optimal. While a variety of such probabilistic ranking approaches have been explored in the generative linguistics literature, within the OT learnability literature, the most well-known example of probabilistic ranking is Stochastic OT, which comes with an associated learning algorithm, the Gradual Learning Algorithm (Boersma 1997; Boersma and Hayes 2001). Section 3.1 presents the Gradual Learning Algorithm for Stochastic OT and its application to learning with full structural descriptions4. The other method of generalizing classic OT, discussed in Section 3.2, involves numerical constraint weighting rather than ranking. The discussion focuses on Harmonic Grammar (Legendre, Miyata, and Smolensky 1990c; Smolensky and Legendre 2006), out of which OT developed, and also discusses probabilistic weighting, a combination of the two extensions to classic OT.

3.1 Probabilistic Ranking

In Stochastic OT (Boersma 1997; Boersma and Hayes 2001), constraints are not strictly ranked on an ordinal scale. Rather, each constraint is associated with a mean ranking value along a continuous scale. Formally, each ranking value represents the mean of a normal distribution, and all constraints’ distributions are assumed to have equal standard deviations, which are generally arbitrarily set to 2. At evaluation time, a selection point is chosen independently from each of the constraints’ distributions, and the numerical ordering of these

---

4 Jarosz 2006a also develops a theory of learning for a probabilistic extension of OT, but since the main focus of that proposal is on learning without full structural descriptions, it is discussed in Section 4.
selection points determines the total ordering of constraints, with higher numerical values corresponding to higher relative ranks. In this way, Stochastic OT defines a probability distribution over total orderings of constraints. The farther apart the ranking values of two constraints are, the higher the probability of a particular relative ranking between them. Conversely, when the ranking values for two constraints are close, each relative ranking has a good chance of being selected. This possibility enables Stochastic OT to model free variation: if two active constraints conflict, different rankings will correspond to different outputs being selected as optimal on different evaluations. This is the main typological consequence of Stochastic OT that differs from classic OT: it predicts that final-state grammars can be variable. In sum, Stochastic OT maintains OT’s evaluation metric for choosing the optimal output form given a ranking: it differs by allowing a single grammar to vary stochastically among different total rankings.

The Gradual Learning Algorithm for Stochastic OT (GLA; Boersma 1997; Boersma and Hayes 2001) is an online, error-driven learner like EDCD and MRCD. It processes one surface form at a time, and learning is triggered when the output generated by the learner does not match the observed output. As in EDCD and MRCD, the learner uses its current grammatical hypothesis to generate an output form for each learning datum it processes. To generate an output, the GLA samples a total ranking from its grammar by randomly choosing selection points for each of the constraints, and uses the resulting total ranking to generate its output for the learning datum. In the case of a mismatch, the algorithm slightly decreases the ranking values of loser-prefering constraints and slightly increases the ranking values of winner-prefering constraints. All constraints are adjusted by the same amount, called the plasticity. To some extent, the update rule for the GLA resembles the update rule for EDCD, except that ranking values of all the loser-prefering constraints are pushed down little-by-little over many errors, and the ranking values for winner-prefering constraints are also adjusted. The basic insight is that as learning continues, constraints favoring losers will gradually be pushed lower and lower - and those favoring winners higher and higher - until errors become diminishingly rare. In general, the algorithm is not guaranteed to converge on a correct grammar, or any grammar for that matter, as shown most concretely by Pater (2008). Pater shows that the GLA can fail to converge on a grammar for a simple language generated from a total ranking of constraints. In practice, however, the algorithm usually performs well assuming it is provided with full structural descriptions of the learning data (see e.g. Boersma 1997; Boersma and Hayes 2001 for simulations).

Probabilistic extensions of OT offer a number of advantages over discrete ranking. The Constraint Demotion family of algorithms discussed in Section 2 assumes that the learning data are free of noise and errors. If presented with data containing errors or variation, the Constraint Demotion family of algorithms will not find a total ranking - it will either detect inconsistency (MRCD) or endlessly cycle between different rankings (EDCD). In contrast, probabilistic extensions of OT enable resistance to noise and errors due to the learner’s sensitivity to frequency. In the GLA, each update to the grammar is small, but grammar updates accumulate over many exposures, and therefore more frequent patterns influence the grammar more over time. This sensitivity to frequency means that the effect on the grammar of a small proportion of errors in the data is overpowered by systematic, high frequency patterns. It is possible to view the noisy evaluation of Stochastic OT solely as a component of the learning process, which is removed upon the completion of learning (see e.g. Boersma and Pater 2008, Section 7). Under this view, the learning task is taken to be learning of a total ranking, and probabilistic learners like the Gradual Learning Algorithm (GLA) have the advantage over Constraint Demotion in being resistant to small amounts of noise. On the other hand, as discussed above, the GLA, in contrast to the Constraint Demotion algorithms, is not provably correct.
In general, however, noisy evaluation is usually taken as more than a noise-resisting component of learning; it is usually taken to imply substantive predictions about typology, distinct from those of Classic OT. Namely, it predicts that adult languages can exhibit optionality or free variation. Since probabilistic extensions of OT extend the range of possible grammars, they extend the grammar-learning subproblem to the learning of these probabilistic grammars. In other words, the grammar-learning subproblem becomes larger (and harder) because the task of the learner now includes not only learning of languages generated from total rankings, but also of languages generated from mixtures of total rankings. Probabilistic learners like the GLA have the ability to match frequencies in the learning data, thereby learning languages with free variation. The GLA has also been used to learn stochastic rankings that can be used to model aspects of gradient grammaticality, where a structure’s grammaticality is modeled in terms of its production frequency under the stochastic grammar (Zuraw 2000; Boersma and Hayes 2001; Hammond 2004; Hayes and Londe 2006). A final advantage of probabilistic grammars, discussed further in Section 6, regards the modeling of acquisition. By representing grammars as distributions over rankings, probabilistic learners can model gradual learning curves, whereby accuracy on individual overt forms improves gradually over time.

3.2 Weighting and Probabilistic Weighting

There are two main types of weighted constraint grammars differing in how the numerically weighted constraints are interpreted at evaluation time. Both evaluate competing output structures based on their relative harmony, which is the weighted sum of constraint violations. The weight of each constraint is multiplied by the number of violations it incurs (expressed as a negative integer), and the results are summed over all constraints as illustrated in Tableau (5).

<table>
<thead>
<tr>
<th>(5) Weighted Grammars and Calculation of Harmony</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lupz/</td>
</tr>
<tr>
<td>a. lupz</td>
</tr>
<tr>
<td>b. lups</td>
</tr>
<tr>
<td>c. lup</td>
</tr>
<tr>
<td>d. lu</td>
</tr>
</tbody>
</table>

Harmony = -25  
Harmony = -20  
Harmony = -22  
Harmony = -44

In Harmonic Grammar (HG) (1990a; 1990b; Legendre et al. 1990c; Smolensky and Legendre 2006) and its close relatives, such as Linear OT (Keller 2000), the optimal output is defined as the output with highest harmony. Thus, HG defines candidate (b) in Tableau (5) as optimal. In a probabilistic extension of HG, called noisy HG (Boersma and Pater 2008), the weights of the constraints are selected from independent normal distributions at evaluation time, just as in Stochastic OT. The difference is that in Stochastic OT these numerical weights are interpreted as a strict ranking, whereas in noisy HG, they correspond directly to the weights used in evaluation. Thus, noisy HG defines a probability distribution over weightings of constraints in the same way that Stochastic OT defines a probability distribution over rankings. This variation in weights/rankings determines the probability with which different output structures are selected as optimal.

In Maximum Entropy (also called log-linear) grammars, which have also been applied to learning with OT-like constraints (Johnson 2002; Goldwater and Johnson 2003; Jäger 2007), the probability associated with an output candidate is directly related to the harmony. Unlike Noisy HG, Maximum Entropy models use a single weighting to define the probability
with which different candidates are selected: specifically, the probability of an output is proportional to the exponential of its harmony. This contrasts with noisy HG, according to which only candidates that win under some weighting are assigned probability. As a result, noisy HG assigns zero probability to candidates like (d) above, while a Maximum Entropy model assigns nonzero probability. In sum, while the stochastic component in noisy HG resides in the weightings themselves being noisy, the stochastic component in Maximum Entropy models exists at the level of candidate output structures directly.

Given full structural descriptions of the data, HG, noisy HG, and Maximum Entropy grammars can all be learned by an online, error-driven learning algorithm very much like the GLA (Jäger 2007; Boersma and Pater 2008)\(^5\). This algorithm for weighted grammars, called Stochastic Gradient Ascent/Descent (SGA), is error-driven: when there is an error, weights of loser-preferring constraints are slightly decreased, and weights of the winner-preferring constraints are slightly increased, just as in the GLA. The only difference between the GLA and SGA update rules is that in SGA the amount of change for the weight is proportional to the difference between the number of constraint violations assigned to the winner and the number of constraint violations assigned to the loser, whereas in the GLA all that matters is which form has more/fewer violations (Jäger 2007; Boersma and Pater 2008). This slight difference in the update rule has important formal consequences: the learning algorithms for HG, noisy HG, and Maximum Entropy grammars are provably convergent on the correct (nonvarying) target grammar given inputs paired with fully structured outputs (Fischer 2005; Jäger 2007; Boersma and Pater 2008).

In sum, weighted grammars of various sorts can all be learned by an online, error-driven learning algorithm very much like the GLA for Stochastic OT, but, unlike the GLA, Stochastic Gradient Ascent is provably correct. Furthermore, probabilistic weighting (noisy HG and Maximum Entropy) shares the advantages of probabilistic ranking with respect to noise tolerance, learning of variation, and prediction of gradual learning curves. Both HG and Maximum Entropy grammars have been used to model gradient grammaticality. Indeed, the original application of HG was to model the interaction of syntactic and semantic factors on the graded acceptability of intransitive sentences in French (Legendre et al. 1990a; 1990c). More recently, modeling of gradient grammaticality has been explored for Maximum Entropy grammars (Hayes and Wilson 2008) as well as for HG (Keller 2000; Coetzee and Pater 2008a)\(^6\).

Despite these many advantages, weighted constraint systems have the potential disadvantage of fundamentally departing from the typological predictions of Classic OT, for which there is extensive empirical support. Indeed, OT developed out of HG, and Prince and Smolensky (2004) dismissed HG as insufficiently restrictive. Recently, there has been a resurgence of interest in weighted grammars, however. For a fixed set of constraints, weighting is a more powerful system that permits a range of additive constraint interactions, or cumulative effects. Some authors have argued that the move to the weighted constraint grammar model is warranted in order to account for attested cumulative effects (Keller 2000; Keller and Asudeh 2002; Goldwater and Johnson 2003; Jäger and Rosenbach 2006; Coetzee and Pater 2008a)\(^7\). Pater (2009) argues that HG is not as powerful as has been assumed, that

\(^5\) See Soderstrom, Mathis, and Smolensky 2006 for a related learning algorithm for a connectionist implementation of HG, and Johnson 2002; Goldwater and Johnson 2003 for other learning algorithms for Maximum Entropy grammars. See also Bane, Riggle, and Sonderegger to appear on general properties of learning with HG, including a linear bound on the number of training examples necessary to learn a weighting of constraints.

\(^6\) Actually, Hayes and Wilson’s model also learns the constraints themselves. For other work on learning constraints see Hayes 1999; Flack 2007.

\(^7\) See (Hayes and Londe 2006; Jäger and Rosenbach 2006; Jarosz 2010), however, on how Stochastic OT is able to account for certain kinds of additive constraint interactions.
Learning with Violable Constraints – Gaja Jarosz

many bizarre predictions go away when certain constraints that are problematic on independent grounds are removed, and that HG more elegantly captures certain cumulative effects than alternative approaches. As Pater points out, it would be a mistake to assume that a typological theory using weighted constraints must rely on the same constraints that OT relies on, and therefore the relative restrictiveness of the two theories is not clearly in a subset/superset relationship. However, Bane and Riggle (to appear) show that even using basic constraints like MAX and DEP gives rise to novel (and arguably undesirable) typological predictions in HG. In sum, the appropriateness of weighted constraint grammars as a typological theory is an open question and a topic of much ongoing debate (Prince 2002b; Legendre, Sorace, and Smolensky 2006; Tesar 2007; Pater 2009; Bane and Riggle to appear; Potts, Pater, Jesney, Bhatt, and Becker to appear).

3.3 Summary

Classic OT ranking can be extended in various ways by replacing discrete ranking with ranks/weights on a continuous scale and/or by adding noise to the evaluation process. Probabilistic variants allow for tolerance of noise and errors in the learning data and for learning of variable target languages. Of the three probabilistic variants discussed above, only the SGA for noisy HG and the SGA for Maximum Entropy grammars are guaranteed to converge on a grammar consistent with the data; the GLA for Stochastic OT does not always converge on a grammar consistent with the data, even for data generated from total rankings of constraints. On the other hand, Stochastic OT is a more restricted generalization of Classic OT, and therefore its typological consequences are closest to that of Classic OT. Weighted grammars generalize Classic OT in an orthogonal direction, abandoning strict domination in favor of additive constraint interaction, thereby predicting a range of cumulative effects. There exist provably correct learning algorithms for both probabilistic and non-probabilistic weighted grammars. The typological consequences of assuming a grammar relying on weighting rather than ranking are a topic of much ongoing debate.

4. Hidden Structure

The presentation so far has focused on online, error-driven learning algorithms for (probabilistic) ranking and weighting for the narrow grammar-learning subproblem. As these sections have shown, there are a number of available learning algorithms that solve this important subproblem. However, the success of all of these learning algorithms relies on the assumption that learners are provided with full structural descriptions of the data, including prosodic/syntactic structure as well as underlying representations, which are not available to the human learner.

To understand the challenge posed by hidden structure, recall that the grammar update for all these error-driven learners involves comparing the constraint violations of the learner’s output and the learning datum in order to determine how the rankings or weightings of constraints should be adjusted. Hidden structure obscures the constraint violations of the observed learning data. Constraint violations are assigned to fully structured input-output pairs, not to unstructured overt forms that the human learner is presumably exposed to. Structural ambiguity, created by prosodic or syntactic structure, obscures the violations of any constraints that reference the hidden structure, whether they assign that structure or simply depend on it. For example, if the learning datum is a tri-syllabic word with medial stress, e.g. [pakæti], there are (at least) two analyses: [(pakæ)ti], with an initial iambic foot, and [pa(kæti)], with a final trochaic foot. Without the footing, violations of constraints like TROCHAIC and IAMBIC for the learning datum are unknown. If underlying/lexical representations are unknown, then the violations of faithfulness constraints for the observed
Learning with Violable Constraints – Gaja Jarosz

output are likewise unknown. Thus, without the full structural descriptions, the vector of constraint violations for the observed output required to calculate the update to the grammar is not apparent from the learning datum.

Learning without access to full structural descriptions is a major and rich area of ongoing research. Work in this area usually focuses on one of two major subproblems: structural ambiguity and the learning of the lexicon. Sections 4.1–4.3 review several learning theories that address the problem posed by structural ambiguity (the broad grammar-learning subproblem) and Section 4.4 briefly reviews work on the learning of the lexicon.

4.1 Robust Interpretive Parsing

Within OT, learning in the face of structural ambiguity has been a topic of ongoing work since at least Tesar (1997a; 1998) and Tesar and Smolensky (1998; 2000). In order to apply error-driven learning in the presence of structural ambiguity, Tesar and Smolensky (1998) proposed Robust Interpretive Parsing (RIP), which provides an educated guess, based on the current constraint ranking, about the structure of the observed output. Specifically, RIP uses the learner’s current hierarchy to select the most harmonic candidate among the structural descriptions consistent with the learning datum. That is, RIP uses standard OT evaluation, but rather than choosing between competing pronunciation of the same input, it chooses between competing analyses, or parses, of the same overt form. RIP thus selects an analysis of each learning datum and a corresponding full structural description that enables the grammar updates to be calculated in the usual way. RIP is motivated independently as a mechanism for analyzing/parsing overt structures given a (adult) grammar; however, in the context of learning, RIP does not always work because the learner’s (incorrect) grammar can cause RIP to assign incorrect structure, leading the learner astray. Tesar and Smolensky (2000) presented simulation results for a RIP version of Constraint Demotion on a large metrical phonology test set with structural ambiguity. They found that RIP/CD learned just 60.5% of the languages in the system correctly when starting from an unranked initial hierarchy.

RIP was later applied to the GLA (Boersma 2003; Apoussidou and Boersma 2003; Apoussidou 2007), as well as to SGA for HG (Boersma and Pater 2008). Boersma and Pater report on simulations comparing the performance of RIP variants of CD, GLA, and SGA for HG with the same test set used by Tesar and Smolensky (2000). They found that RIP/SGA and RIP/GLA outperform RIP/CD, with RIP/SGA for noisy HG getting the highest performance, learning almost 89% of the languages in the system on average. These results appear promising for RIP/SGA, which dramatically outperforms RIP/CD and RIP/GLA. However, Jarosz (to appear) showed that a baseline algorithm, relying on random search, outperforms all of these RIP-based algorithms on the same test set, given the same learning conditions. Indeed, the random baseline algorithm learned all the languages on all learning trials, getting 100% accuracy. Measuring performance of explicit baseline models is important because it quantifies the difficulty of the learner’s task and the efficiency of learning algorithms on that task. As Jarosz (to appear) shows, the difficulty of the learning task is difficult to anticipate. These results mean that it is vital to evaluate these and other learning algorithms on much larger, harder problems to ensure that their performance and efficiency will scale better than random search.

4.2 Inconsistency Detection

In another strand of research, Tesar (1997b; 2004b) develops an approach to structural ambiguity that builds on MRCD’s ability to detect inconsistency. Recall that the learner’s

---

8 Boersma and Pater 2008 show that the accuracy of RIP/CD falls to around 47% on this test set when permuting ties rather than pooling ties are used.
Learning with Violable Constraints – Gaja Jarosz

hypothesis maintained by MRCD is actually a list of winner-loser pairs, the support, which is consulted during the processing of each learning datum. In order to deal with structural ambiguity, the Inconsistency Detection Learner (IDL) maintains a set of supports, each corresponding to an internally consistent combination of analyses of the learning data examined so far. For each learning datum, IDL runs RCD on each of supports to determine if the ranking for that support generates an error for that datum. In order to learn from the error, the algorithm needs to assign a structural description to the learning datum. IDL does so by considering each analysis of the datum in turn. Specifically, for each support that yields an error, IDL constructs a version of that support for each structural analysis of the current learning datum and tests each resulting support for consistency using RCD. Supports that are inconsistent are discarded, while supports that are consistent are retained for further processing. In this way, the learner is guaranteed to find a combination of analyses, and an associated ranking, that is consistent with all the data, assuming the data were generated from a total ranking of constraints. Unlike MRCD, IDL does not come with a general proof of its efficiency: the efficiency of the algorithm depends on the degree of structural ambiguity present in the data and on the extent to which the learning data are mutually constraining. However, based on simulations with metrical phonology systems like those of Tesar and Smolensky (2000), Tesar (1997b; 2004b) showed that the total number of supports that IDL considers during learning for such systems is relatively small because many supports can be quickly ruled out due to inconsistency.

In sum, IDL is guaranteed to find a consistent hypothesis, but it does involve considerable additional complexity beyond that of MRCD, GLA, and SGA, which maintain just one grammatical hypothesis at a time. It is also important to note that the inconsistency detection approach is incompatible with noisy or variable data since the approach searches for a hypothesis consistent with all the data.

4.3 Likelihood Maximization

Jarosz (2006a; 2006b) proposes a theory of constraint-based learning that addresses the full problem of learning both the grammar and the lexicon of underlying forms given unstructured surface forms. Maximum Likelihood Learning of Lexicons and Grammars (MLG; Jarosz 2006a) formalizes learning as an optimization problem within the general framework of likelihood maximization. MLG differs fundamentally from the error-driven learning approaches discussed above and does not rely on access to the full structural descriptions of the learning data. Although MLG provides a unified solution to hidden structure of both kinds, the present discussion focuses on structural ambiguity, and the learning of the lexicon in MLG is discussed in the next section.

MLG assumes that a grammar is defined as a probability distribution over rankings, as in Stochastic OT. Learning in MLG is not error-driven, however. Instead, for each datum, the learner determines what components (or parameters) of the grammar are capable of generating the datum, and it rewards those components. Simplifying somewhat, for each datum the learner determines which rankings can generate the datum, and it rewards successful rankings. More precisely, the learner rewards a ranking in proportion to its probability given the datum and the current grammar. Crucially, in order to determine whether a component of the grammar is able to generate the datum, the learner need only determine whether the overt portion of the learner’s output matches the overt portion of the

---

9 In the general case the algorithm must iterate through all the structural analyses of a datum for which an error is produced one-by-one. As noted by Eisner 2000, this set could be very large (such as with the case of syntactic parse trees) - this raises questions about the tractability of this procedure in the general case.

10 However, given that mutually constraining characteristics of the system may also benefit the baseline learner, Jarosz (to appear) advocates that IDL’s efficiency be explicitly compared to that of the baseline.
learning datum. The full structural description of the learning datum is irrelevant for determining how much a grammar component should be rewarded - only the overt form is required. In this way, updates in MLG do not depend on assigning a structural interpretation to each overt form. Since optimization is defined over probabilistic grammars, MLG exhibits a sensitivity to frequency and robustness to noise, like the other probabilistic approaches (Jarosz 2006a; Jarosz 2006b). In addition, formalization of the learning problem as an optimization problem makes available a wealth of well-understood and mathematically sound techniques for performing the optimization. Jarosz (2006a) illustrates the capacities of the theory to learn structurally ambiguous syllabification using the well-known Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977), adapted appropriately to the task. EM makes gradual updates to the grammar and is guaranteed to converge on a (local) maximum.

The challenge for this approach is the identification of an appropriate representation of the grammar that allows updates to the grammar to be made efficiently. The simulations presented by Jarosz (2006a) made the simplifying assumption that the grammar is represented as a probability distribution over the set of total rankings, making the procedure intractable for larger constraint systems. In recent work, Jarosz develops a variant of the theory, and a corresponding family of learning algorithms, that do not make these simplifying assumptions (Jarosz 2009). The family of algorithms includes an online, sampling variant of EM that maintains a single grammar hypothesis at a time, represented in terms of a stochastic extension of partial ordering. Whereas true EM updates rely on calculating expectation over the entire data in batch, the basic idea behind these sampling variants is that for a given overt form, the learner uses the current grammar to randomly sample a ranking compatible with it, and then rewards that ranking. Thus, whereas in error-driven learning the goal of processing each datum is identifying losing competitors, in MLG the goal is identifying successful rankings to reward. Jarosz (2009a) shows that this family of algorithms achieves higher accuracy than RIP/SGA on Tesar and Smolensky’s (2000) test set given the same learning conditions but does not compare their performance to the baseline discussed in Jarosz (to appear).

4.4 Learning Underlying Forms

The previous sections discussed learning models that address the broader grammar-learning subproblem, an area of much ongoing work. Another area of ongoing research, and one that is less well understood, is the simultaneous learning of grammars and lexicons. The three different approaches to the broader grammar-learning subproblem - RIP, IDL, and Likelihood Maximization – are also the three main lines of attack on the problem of learning underlying representations. After considering the challenge posed by this problem, this section briefly reviews these approaches.

The simultaneous learning of a grammar and a lexicon is a particularly challenging task due to the interdependence of these two components of linguistic knowledge. The models discussed so far have assumed the lexicon is available to the learner. In general, however, the choice of lexicon depends on the grammar, and the choice of grammar depends on the lexicon (Tesar and Smolensky 2000; Albright and Hayes 2002; Tesar et al. 2003; Tesar 2006a). For example, an alternation like [rat] ~ [radə] can be accounted for by a lexical entry /rad/ with final devoicing or a lexical entry /rat/ with intervocalic voicing. How can the learner make any decisions about either component without knowing anything about the

---

11 Tesar and Smolensky 1998; 2000 proposed lexicon optimization as a solution to the problem of learning a lexicon given an adult grammar.
other? From the perspective of an error-driven learner, when an error is produced, it is not clear whether the problem should be attributed to the grammar, the lexicon, or both.\(^\text{12}\)

One major approach to the learning of underlying representations builds on MRCD and inconsistency detection (Tesar et al. 2003; Prince and Tesar 2004; Tesar 2004a; Alderete, Brasoveanu, Merchant, Prince, and Tesar 2005; Tesar 2006a; Tesar 2006b; Merchant 2008; Merchant and Tesar 2008; Tesar 2008; Tesar 2009; Akers 2011). This approach relies on the mutually constraining nature of grammatical and lexical information. As the learner builds up some information about the grammar, this grammatical information allows some aspects of the lexicon to be inferred, which in turn allows further inferences about the grammar, with the process iterating between grammar and lexicon learning. Inconsistency detection plays a key role in the inference of grammatical as well as lexical information. An important technique within this approach is *Contrast Analysis*, which considers pairs of surface forms differing in exactly one morpheme (Tesar 2004a; Alderete et al. 2005; Tesar 2006a; Tesar 2006b; Merchant 2008; Merchant and Tesar 2008). The learner uses such pairs to test possible underlying features for a morpheme against the amassed grammatical information. When a certain feature value leads to inconsistency, the learner can safely set the feature to the opposing value. Recent work has improved on the efficiency of such inferences by assuming certain restrictions on phonological grammars (Tesar 2008; Tesar 2009; Akers 2011).

The Likelihood Maximization approach to learning lexicons in MLG (Jarosz 2006a) is the same as the approach to dealing with structural ambiguity: rewarding of successful lexical representations depends only on matches with the overt data. MLG assumes a probabilistic extension of the standard generative phonology model: the learner selects an underlying representation for a given morphological/semantic input, and then uses that underlying representation and the grammar to generate a surface structure. The lexical representation in MLG is probabilistic, however, which means that each morpheme is associated with a distribution over possible underlying representations, and this distribution changes gradually during learning. During simultaneous grammar and lexicon learning, the learner rewards both lexical and grammatical representations that are successful in generating the overt data (Jarosz 2006a; Jarosz 2009; Jarosz 2011).

A third approach to learning underlying representations assumes that the grammar and lexicon interact in parallel, with the lexicon represented in terms of *lexical constraints* (Apoussidou 2006; Apoussidou 2007). Lexical constraints connect abstract morphemes/meanings with possible underlying representations: there is one constraint for each possible underlying representation of each morpheme, and these constraints interact in parallel with standard grammatical constraints. As a result of this novel representation of the lexicon, lexical learning can be viewed as a constraint ranking problem, just like grammar learning. However, as with structural ambiguity, the learner cannot directly observe the constraint violations of the observed data because the underlying representations associated with overt learning data are hidden. Apoussidou proposes to deal with this by using what is essentially RIP/GLA (see Section 4.1 above) except that interpretive parsing consists of selecting an underlying form, rather than the structure, consistent with the overt portion of the learning datum. The learner generates its own underlying representation and surface form for

\(^{12}\) Actually, the challenge is greater than this – the learner must not only (efficiently) find a lexicon and grammar that together generate the learning data, the learner must identify a restrictive grammar. The problem of learning restrictive languages is discussed in Section 5 – in the context of simultaneously learning a grammar and lexicon, this problem is particularly relevant because systematic restrictions in the target language can be (incorrectly) accounted for by making restrictions on the lexicon. Thus, the learner must not only identify a restrictive grammar and lexicon combination, the learner must ensure that all systematic restrictions are handled by the grammar. See Jarosz 2006a; Tesar 2006a; Jarosz 2009 for further discussion of this challenging issue.
Learning with Violable Constraints – Gaja Jarosz

the learning datum and compares it to the full structural description of the learning datum provided by robust interpretive parsing. If the underlying representations and overt forms do not match (an error), the GLA update rule can be applied as usual to adjust the ranking values of the constraints, which in this case include both lexical and grammatical constraints.

4.5 Discussion

This section has reviewed several developing approaches for dealing with two kinds of hidden structure, structural ambiguity and underlying representation. In many cases, work on one problem has assumed a solution to the other, but in general, grammars and lexicons must both be learned from (potentially) structurally ambiguous overt forms. All three of the main approaches have been applied to this full learning problem (Jarosz 2006a; Apoussidou 2007; Akers 2011), but this is an area where much further work is needed.

5. Learning Restrictive Languages

Identifying learning models that are capable of learning combinations of lexicons and grammars in an efficient and psychologically plausible way from unstructured overt forms is a major challenge and the focus of much ongoing work, as discussed above. This, however, is still not the full problem facing the language learner. In general, there are many grammars that are consistent with a set of learning data, and the learner’s task is to identify the most restrictive grammar among these, the one that is able to generate all the observed forms and as few additional forms as possible. Restrictiveness entails that the learner’s grammar does not overgenerate by accepting forms that are ungrammatical in the target language. For example, if the learner has encountered no syllables with codas, and yet no alternations that eliminate or avoid codas, the learning data would be consistent with either a high or a low ranking of NOCODA. However, it is generally assumed that the learner should in this case acquire a grammar with high-ranked NOCODA so that syllables with codas would be rejected as ungrammatical.

Within OT, a well-known solution to the restrictiveness problem relies on ranking biases, the most well-known of which is the Markedness » Faithfulness (M » F) bias (Smolensky 1996). In general, ranking markedness constraints as high as possible favors grammars that avoid marked configurations by mapping them to unmarked alternatives. The M » F bias does not exhaust the notion of restrictiveness, however; additional ranking biases are needed to supplement the M » F bias. These include the Specific-Faithfulness » General-Faithfulness bias (Smith 2000; Hayes 2004; Tessier 2007) and the Output-Output Faithfulness » Input-Output Faithfulness bias (McCarty 1998; Hayes 2004; Tessier 2007).

One simple way to incorporate ranking biases in learning models is to assume the biases determine the initial state of learning (Smolensky 1996; Boersma and Leveit 2000; Hayes 2004; Jesney and Tessier to appear). In this way the learner begins with a restrictive grammar and modifies it in response to overt data. Implementing a M » F bias is straightforward since constraints can easily be identified as Markedness or Faithfulness constraints. Implementing the Specific-Faithfulness » General-Faithfulness bias is trickier, however, since the general-to-specific relationships between constraints are language-specific and must themselves be learned (Prince and Tesar 2004; Tessier 2007). Furthermore, Prince and Tesar (2004) and Hayes (2004) show that starting from a biased initial ranking is not sufficient to ensure that learners end up in a grammar respecting the bias. Prince and Tesar (2004) and Hayes (2004) independently develop learning algorithms based on MRCG, called Biased Constraint Demotion (BCD) and Low Faithfulness Constraint Demotion (LFCD), respectively, which incorporate persistent ranking biases that affect ranking decisions throughout the learning process. As Prince and Tesar explain, guaranteeing that markedness
constraints end up as high as possible is not trivial, however, because the choice of which faithfulness constraint to rank (when one must be ranked) can affect the learner’s ability to rank markedness constraints highly later on. Subsequent work has argued that implementing ranking biases in the initial stage of learning is more effective for GLA learners than for CD learners (Boersma and Levelt 2003) and more effective for SGA/HG learners than for GLA learners (Jesney and Tessier to appear). In sum, ranking biases provide an indirect way to approximate the restrictiveness of a grammar, and their implementations and effectiveness varies across learning models.

In addition to exploring the effects of different implementations of ranking biases, one area for further work is the concurrent learning of restrictive grammars and lexicons. In general, the restrictiveness of grammar and lexicon combinations cannot be reduced to the relative ranking of particular classes of constraints (Alderete and Tesar 2002; Jarosz 2009). The restrictiveness of a learner’s hypothesis depends crucially on the lexicon, and the choice of the lexicon can determine the restrictiveness of the hypothesis as a whole (Alderete and Tesar 2002; McCarthy 2005; Jarosz 2009). For example (Alderete and Tesar 2002), consider a language that has regular final stress unless the final vowel is epenthetic, in which case stress is penultimate. The learner can explain the penultimate stress forms in one of two ways, either by assuming a lexical stress difference and faithfulness to underlying stress, or by assuming epenthesis and avoidance of stress on epenthetic vowels. The problem is that both solutions involve high ranking of a faithfulness constraint and cannot be distinguished by ranking biases. Nonetheless, the solution relying on faithfulness to underlying stress overgenerates, predicting penultimate stress for forms with any final vowel, not just an epenthetic one. One general approach to dealing with the restrictiveness of grammar and lexicon combinations is developed within MLG. Specifically, in MLG the restrictiveness of grammars is defined directly in terms of the ability of the grammar to maximize the likelihood of the learning data given a rich base (Jarosz 2006a; Jarosz 2006b). The rich base is simply the universal space of possible underlying forms, which is independently needed as the source of possible underlying forms for lexical learning. Jarosz (2009b) shows that this formulation is more general than any of the ranking biases and extends to subset/superset relationships that arise due to the interaction of grammars and lexicons. In general, further work in this area is needed to better understand the scope of the problem and the extent to which existing approaches can cope with the kinds of restrictiveness issues that arise during the simultaneous learning of lexicons and grammars.

6. **Modeling Acquisition**

There is a rich literature on formal modeling of acquisition with violable constraints and a good deal of learnability work that has aimed to model these results computationally. Developmental research has identified a number of general characteristics of child language, characteristics that differentiate child grammars from adult grammars and which computational modeling work has sought to capture. It has been observed that child grammars are generally less marked than adult grammars and that acquisition involves a progression from an unmarked initial state to the target grammar via the gradual addition of successively more marked structures (Jakobson 1941 / 1968; Stampe 1969). Within constraint-based frameworks, such a progression can be explained by positing an initial ranking of \( M \succ F \) (Smolensky 1996; Gnanadesikan 1995 / 2004), the same bias that is used in natural language acquisition.

---

13 See Tessier 2009, however, on hypothetical learning situations arising during RIP-type learning that could lead learners into superset traps that she suggests would be more difficult for memory-less learners like GLA and SGA to get out of. Further investigation of these issues is needed to understand the conditions under which such scenarios could arise and the capacities of various learners to deal with them.
independently motivated to favor the learning of restrictive final-state grammars. Assuming an initial ranking of M » F has become standard practice in work on computational modeling of acquisition. Another characteristic of child grammars is their variable or noisy nature as compared to adult grammars (see e.g. Legendre, Hagstrom, Vainikka, and Todorova 2002). Children’s production is notoriously variable throughout the process of acquisition, with pronunciations of a single word or structure varying from one utterance to another.

One aspect of gradual learning that has been examined in several studies concerns the interacting roles of markedness and frequency and their effects on the order of acquisition. Based on the study of twelve Dutch children’s development of syllable structure, Levelt, Schiller, and Levelt (2000) and Levelt and van de Vijver (1998 / 2004) found that the acquisition orders of all twelve children were consistent with a markedness bias. However, they also found an effect of frequency, with higher frequency syllable types being acquired earlier. They proposed specific roles for universal markedness and language-particular frequency, with markedness determining possible orders cross-linguistically and frequency playing a secondary role, determining acquisition order for structures not differentiated by universal markedness. In subsequent work, Boersma and Levelt (2000) showed that the GLA was able to reproduce these acquisition orders given an initial M » F grammar and the frequencies of various syllable types in Dutch. Recently, Jarosz (2010) extended these results to other languages as well as to other probabilistic constraint-based learners. Specifically, Jarosz found that attested relative orders of acquisition of complex clusters in English, German, French, and Polish were consistent with the kind of markedness and frequency interaction proposed by Levelt and colleagues. Furthermore, she showed via simulations with the GLA, SGA for noisy HG, and MLG, that all three of these models were able to predict the distinct acquisition orders across languages with distinct frequencies of syllable types.

Several additional predictions of these probabilistic learning models are important to note. Due to the incremental nature of learning and the fact that grammars are probability distributions over rankings/weighting, these models predict gradual learning curves with gradual learning of individual target forms and variation throughout the learning process. As production on a particular target form gradually improves over time, the learner passes through intermediate grammars exhibiting variation for the target form, where the target form is sometimes produced accurately and other times reduced to a less marked structure. The rate of accurate production of the target form improves gradually during learning, and the learner eventually settles on a grammar that consistently generates the target form. Thus, a natural consequence of gradual learning with probabilistic grammars is the ability to predict fine-grained learning curves as well as intermediate grammars characterized by variation.

Tessier (2007; 2009) develops an extension of MRCD that models acquisition as gradual re-ranking in a Classic OT grammar. As Tessier explains, RCD alone cannot model gradual re-ranking because it identifies a ranking that is compatible with all winner-loser pairs in a single application. In Error-Selective Learning (ESL) the learner stores errors in a temporary structure called the Error Cache and gradually adds errors to the support in such a way that re-ranking of the current hierarchy is minimal. This results in a series of classic OT grammars that are intended to model intermediate acquisition stages (as well as initial and final stages). Modeling acquisition in terms of a sequence of rankings predicts discrete stages of acquisition permitting more and more of the target forms to be captured, but it does not allow for the finer-grained modeling possible with probabilistic grammars, which can also

---

14 Another characteristic of child language that has been the focus of recent work is the extent to which child grammars exhibit cumulative effects not found in adult grammars (Jarosz 2010; Albright, Magri, and Michaels to appear; Jesney and Tessier to appear).

15 It would be possible for MRCD to model gradual re-ranking if it happened to process the learning data in just the right order. Tessier’s proposal in essence ensures that this happens in general.
model gradual learning of *individual* target forms. This is because in ESL, each stage is a classic OT grammar, and any individual target form is acquired in one step when the markedness constraint it violates is demoted. ESL is guided by frequency in the data in an indirect way: re-ranking is triggered by markedness constraints that are involved in generating sufficiently many errors. Therefore ESL also predicts a general effect of frequency on the order of acquisition, but see Tessier on how the approach differs from the sensitivity to frequency exhibited by probabilistic learners.

An important aspect of ESL is that it incorporates persistent ranking biases, building on BCD (Prince and Tesar 2004) and LFCD (Hayes 2004). Each time the ranking is updated (when an error is added to the support), the learner identifies a ranking that is compatible with the support while at the same time enforcing the ranking biases as much as possible. This means that the learner selects errors that cause minimal change and also allow for the intermediate grammar to be restrictive, reflecting the ranking biases. Tessier shows how such restrictive intermediate stages are necessary in order to capture certain intermediate child grammars, called Intermediate Faith (IF) stages. In IF stages, children’s grammars permit target marked structures only in privileged positions, such as stressed syllables. Tessier shows that while the GLA cannot pass through such stages, the SGA for HG *can* pass through such stages purely as a consequence of the differences in constraint interaction (Jesney and Tessier to appear).

In sum, probabilistic learners like the GLA, SGA, and MLG have a number of advantages with respect to modeling of acquisition. Starting from an initial M » F grammar, they can automatically derive order of acquisition effects as a consequence of their sensitivity to frequency. They also produce detailed and gradual learning curves and correctly predict intermediate grammars to be highly variable. Acquisition order can also be modeled as a sequence of total rankings by the ESL learner. Via the incorporation of persistent ranking biases, ESL is able to model intermediate stages that the GLA with just a M » F bias cannot. For the most part, modeling of acquisition has focused on modeling aspects of the narrow grammar-learning subproblem. There is much work left on comparing the predictions of the various models and connecting their predictions to developmental findings across linguistic domains. An important issue for further work is determining how the simultaneous learning of a grammar and lexicon and the presence of structural ambiguity affect the predictions of computational models for acquisition. There is much potential for productive collaborative efforts between researchers in developmental linguistics and those working on computational modeling. Developmental findings can inform the design and evaluation of computational models; conversely, existing computational models can be used to generate concrete predictions of learning theories for further experimental or developmental investigation.

7. **Conclusion**

Computational modeling of learning for constraint-based grammars is an extremely rich area of research. For each learnability subproblem there now exist several alternative computational models with different strengths and weaknesses whose predictions and differences are being explored in ongoing work. Significant progress on each of the learnability subproblems has already been made, furthering our understanding of how the difficult task of language learning can be solved by language learners. The concrete connections between the predictions of computational models and developmental findings

---

16 However, see (Jarosz 2011) for simulations with MLG modeling experimental results on the simultaneous learning of the grammar and lexicon and Tessier 2009 for discussion of modeling acquisition of the broader grammar-learning subproblem.
provide further, independent support for existing constraint-based learning models, and further our understanding of how children learn language.

Nonetheless, many challenges remain for future work. The capacities of various learning models to deal with the narrow grammar-learning problem are relatively well understood; indeed, several of the existing models have proofs of correctness and/or efficiency. The capacities of learning models to deal with the broader grammar-learning problem and the simultaneous learning of lexicons are less well understood. Much further work will be needed to explore the limits of successful learning, as well as the learning of restrictive grammars, in this context. Another important area for future work is developing richer and more extensive data sets with hidden structure and variation on which the various models can be evaluated. With such a rich array of available approaches, there is much potential for future work to build on the strengths of existing approaches to identify a psychologically plausible model that can learn both restrictive grammars and lexicons from unstructured data.

Author’s Address

Gaja Jarosz
Department of Linguistics
Yale University
370 Temple St., room 204
New Haven, CT 06520
gaja.jarosz@yale.edu

References

Alderete, John & Bruce Tesar (2002), Learning Covert Phonological Interaction: An Analysis of the Problem Posed by the Interaction of Stress and Epenthesis, Ms. Rutgers University, New Brunswick, NJ.
Learning with Violable Constraints – Gaja Jarosz


Bane, Max, Jason Riggle & Morgan Sonderegger (To appear), The VC Dimension of Constraint-Based Grammars. To appear in Lingua.


Boersma, Paul & Joe Pater (2008), Convergence Properties of a Gradual Learning Algorithm for Harmonic Grammar, Ms. University of Amsterdam and University of Massachusetts, Amherst.


Learning with Violable Constraints – Gaja Jarosz


Learning with Violable Constraints – Gaja Jarosz

Jarosz, Gaja (2009), Learning Phonology with Stochastic Partial Orders. In Third North East Computational Phonology Meeting. MIT, Cambridge, MA.


Learning with Violable Constraints – Gaja Jarosz

Tesar, Bruce (1997b), Multi-Recursive Constraint Demotion, Ms. Rutgers University, New Brunswick, NJ.
Tesar, Bruce (2004a), Contrast Analysis in Phonological Learning, Ms. Rutgers University, NJ.
Tesar, Bruce (2007), A Comparison of Lexicographic and Linear Numeric Optimization Using Ldots, Ms. Rutgers University, New Brunswick, N.J.
Tesar, Bruce (2008), Output-Driven Maps. Ms. Rutgers University, New Brunswick, NJ.