Learning syntactic parameters gradually and without triggers

1. Introduction

Since Chomsky (1981), parametric approaches have been widely used in syntax, yet learning syntactic parameter settings has received insufficient attention (though see Yang 2002, Gibson and Wexler 1994, Gould 2015). In particular, little work has been done on the acquisition of realistically-sized parameter systems like those often studied in phonological modeling (e.g. Dresher and Kaye 1990, Tesar & Smolensky 2000, Nazarov and Jarosz 2017, 2019). Our paper represents initial results in that direction: we adapt two domain-general learning algorithms that have been successfully applied to learning of large constraint and parametric systems in the phonological domain, and we successfully apply them to two sets of simple syntactic systems. These algorithms, Expectation Driven Learning (EDL) and Stochastic Gradient Descent (SGD), mark a significant improvement over the existing approaches of Gibson and Wexler (1994) and Yang (2002) in both their ability to learn the correct parameter settings and the gradual way in which they do so. G&W proposed a domain-specific learner which searches for structures that "trigger" certain parameter settings. While this learner acquired the majority of languages in a 3-parameter system, it was unable to capture the full typology. In contrast, the domain-general algorithms EDL and SGD learn this typology without using triggers, and perform equally well when extended to four parameters. Moreover, our models are an improvement over an existing domain-general algorithm, the Naïve Parameter Learner (NPL; Yang 2002), as they learn more effectively and produce gradual learning curves.

2. Learning tasks

The learners were tested on a typology of verb second (V2) movement (see Vikner (1995) for an overview of Germanic V2). This problem provides an instance of the structurally ambiguous data that human learners are often faced with. For example, given a sentence of the form S(ubject) V(erb) O(bject), a learner must determine whether they are in an SVO language, or an SOV language with verb movement to second position. The learner must induce the "hidden structure" (see e.g. Tesar 1998, Jarosz 2006, Pater et al. 2012, Nazarov 2016) - in this case, the underlying linear order and movement operations - that correctly predicts the surface linear order. Our model features inputs containing S, {A(dverb)}, V, and {O}, where {} indicates optionality. A universal c-command hierarchy is assumed for underlying representations: A > S > [V, O]. Following G&W, the 3-parameter task consists of parameters *Comp*, *Spec*, and *V2*. The settings of *Comp* and *Spec* dictate the underlying linear order of the object (relative to V) and subject (relative to [V, O]) respectively. *V2* determines whether the verb fronts to second position; this parameter has the potential to obscure the settings of *Comp* and *Spec*. The 4-parameter task further differentiates between *V2* movement in matrix and embedded clauses by splitting *V2* into two parameters sensitive to clause type.

3. Expectation-Driven and Naïve Parameter learning parametric models

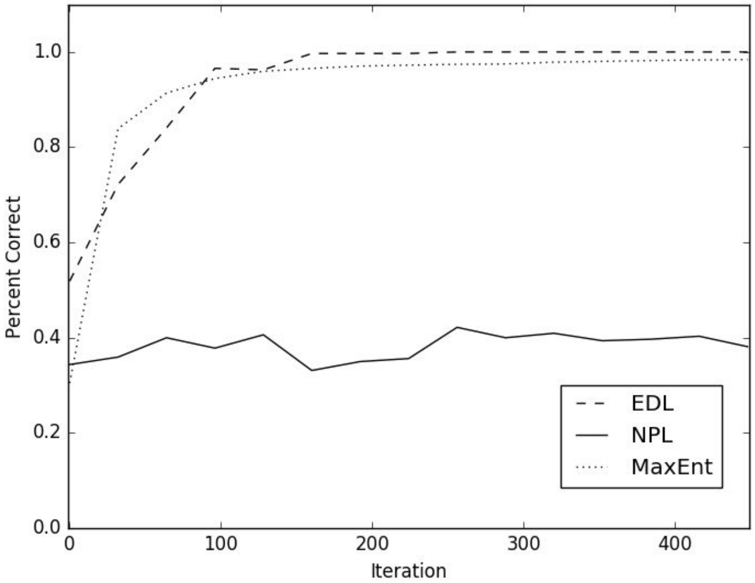
NPL works with a grammar that defines a probability distribution over possible values of a set of binary parameters. When a learning datum is presented, the learner samples a setting for each parameter from its current (probabilistic) grammar. If the sample matches the datum, the probabilities of the current parameter settings are increased, and if it does not, those probabilities are decreased. Blame for the mismatch is attributed uniformly by NPL, with every parameter setting involved penalized when an error occurs. EDL (Jarosz 2015, Nazarov & Jarosz 2017) also uses a probabilistic parametric grammar, but instead assigns blame in proportion to each parameter setting's contribution to the errors (computed using Bayes' Rule). Both algorithms process data online with updates after each observation. Nazarov and Jarosz (2017, 2019) showed that EDL learned parameter values for phonological stress patterns in a way that substantially outperformed NPL.

4. Stochastic gradient descent with a Maximum Entropy model

SGD is an online error-driven algorithm that finds the optimal weights for a set of constraints by estimating how much each constraint contributes to the model's errors. Constraints that cause a model to make incorrect predictions are given smaller weights, while constraints that help the model make correct predictions are given larger ones. We used SGD with a Maximum Entropy (MaxEnt) grammar, which defines a probability distribution over possible output candidates, given an input and a set of weighted constraints (Goldwater and Johnson, 2003). We adapted the MaxEnt model to the parametric learning task by representing each parameter with two "constraints": one violated when the parameter value would be 0, and one violated when the value would be 1. Highly weighted constraints translate to a high degree of confidence in the parameter settings that the constraints correspond to. For calculating online updates in the face of hidden structure, we used Expectation Maximization (Dempster et al., 1977) to compute expected probabilities and constraint violations of the structural descriptions compatible with the observed surface form. This approach adapts Expected Interpretive Parsing (Jarosz 2013) to the syntactic parameter setting and MaxEnt (using exact summation rather than sampling). It is also closely related to the optimization approach used by Pater et al. (2012).

5. Results

In all simulations, the learning rate for EDL and NPL were set to the same value (0.05). All models tested (EDL, NPL, and SGD) were able to successfully learn the 3-parameter typology. This shows that triggers are not necessary to perform the learning task laid out by G&W. The models were then tested on the 4-parameter typology. Results from a representative language (averaged over 10 runs) are shown below:



Both the SGD and EDL models achieved near-perfect accuracy within 200 iterations (where an iteration represents an update of the parameter values/weights after a single learning datum) and did so gradually with accuracy increasing over the course of learning. NPL, however, had some difficulty with this more complex typology. While it eventually converged for all of the languages, in some cases it took thousands of parameter updates to do so. Additionally, NPL's learning was not gradual: the learner's overall accuracy vacillated between relatively high and low accuracies, before increasing dramatically when the algorithm happened upon the correct combination of settings.

6. Conclusions

Here we have taken steps toward modeling syntactic learning using learning models with demonstrated potential (in the phonological domain) to scale-up to more complex datasets. Specifically, we showed that SGD, EDL, and NPL are sufficient for learning the settings for parametric grammars in the cases tested so far, and that these domain-general algorithms all outperformed the trigger-based approach used by Gibson and Wexler (1994). Furthermore, we demonstrated that SGD and EDL outperform NPL in both time-to-convergence and the manner in which they increased their accuracy.