Remarks and Replies

Learning from Triggers

Robert C. Berwick
Partha Niyogi

In this article we provide a refined analysis of learning in finite parameter spaces using the Triggering Learning Algorithm (TLA) of Gibson and Wexler (1994). We show that the behavior of the TLA can be modeled exactly as a Markov chain. This Markov model allows us to (1) describe formally the conditions for learnability in such spaces, (2) uncover problematic states in addition to the local maxima described by Gibson and Wexler, and (3) characterize convergence times for the learning algorithms quantitatively. In addition, we present arguments questioning the psychological plausibility of the TLA as a learning algorithm.

Keywords: principles and parameters, learnability, Triggering Learning Algorithm, Markov chains, local maxima, closed states

1 Introduction

Gibson and Wexler (1994; henceforth, G&W) take important steps toward formalizing the notion of language acquisition in a space whose grammars are characterized by a finite number of *parameters*. One of their aims is to formalize and thereby completely characterize learnability in such spaces, using what they call the "Triggering Learning Algorithm" (TLA). For example, they demonstrate that even in such finite spaces, convergence via some sequence of positive examples ("triggers") may remain a problem, since it is still possible that under a single-step acquisition algorithm the learner can get stuck in a nontarget state. They then investigate several ways to avoid this "local maxima" problem, including some possibilities with linguistic consequences, such as default settings for parameter values (set first −V2; i.e., set the verb-second parameter "off") and "maturation" of parameter settings.

We would like to thank Noam Chomsky, Janet Fodor, Ken Wexler, Ted Gibson, an anonymous reviewer for the Association for Computational Linguistics, and two anonymous *LI* reviewers for valuable discussions and comments on this work; all residual errors are ours. This research is supported by NSF grant 9217041-ASC and by ARPA under the HPCC program on High Performance Computing for Learning.

Linguistic Inquiry, Volume 27, Number 4, Fall 1996
605–622

In this article we will refine G&W's work by providing a more complete and correct picture of learnability in finite parameter spaces. In particular:

1. G&W's algorithm does not completely enumerate the full list of initial and final target grammars for which their learner fails to converge to a target grammar with probability 1. That is, the algorithm in G&W's appendix A does not completely characterize the learnability properties of their example parameter space, or of finite parameter spaces generally. It is not the case that a TLA learner will converge with probability 1 from every (initial state, final state) grammar pair that G&W list as learnable. In fact, out of 56 initial-final grammar state pairs, exactly 12 (as opposed to only the 6 that G&W list in their table 4, p. 426) are not learnable with probability 1. Importantly, the complete list includes some −V2 initial states, contrary to G&W's list in their table 4; see our table 1.
2. We can identify the origin of the flaw in G&W's algorithm. G&W attempt to determine all initial states from which the learner will not converge to the target grammar (with probability 1). Let us call these *problem states*. G&W do this by computing all grammar states *unconnected* to the target grammar by some chain of positive examples (triggers). However, the correct way to find problem states is to compute those states *connected* to *nontarget* local maxima.[1] Crucially, these two ways of computing problem states are not equivalent, because some states might be connected both to the target and to nontarget local maxima (see our figure 1). By G&W's analysis, such states are not problem states; however, by our analysis, they are problem states, as we will show. For instance, in G&W's three-parameter example system (see their table 3), grammar 3 (Spec-final, Comp-first, −V2) is not listed as a problem state; yet starting at grammar 3, a TLA learner will not converge to target grammar 5 (Spec-first, Comp-final, −V2) (with probability 1). We show where in appendix A the algorithm goes awry—in step 3—and provide a correct version for G&W's three-parameter system.
3. More constructively, we provide (as G&W note in their footnote 11) a more precise mathematical formulation of learnability in finite parameter spaces, as a Markov process, that avoids the pitfalls that G&W encounter. Appendices A and B of this article provide the explicit construction. The Markov formalization yields several benefits:
   - For the first time it becomes possible to measure the poverty of the stimulus *exactly*, in the sense that the (average) number of positive stimuli required to reach a target grammar can actually be counted. This can be used as a tool for measuring whether proposed learning algorithms and grammar space parameterizations lead to psychologically plausible convergence times. For example, assuming a uniform distribution over G&W's input sentences, 35 (positive) examples are required on average to converge in their three-parameter system.[2] (It should also be possible to measure the average

[1] As we discuss below, we put to one side the question of possible cycles in the grammar space.
[2] The standard deviation is 22.

number of grammar or hypothesis changes before convergence, but this work remains to be completed.)

- It can also be determined whether proposed maturational solutions are psychologically plausible in the following sense. G&W propose to solve the local maxima problem in their three-parameter space by setting the V2 parameter to the default value −V2 and forbidding the learner to change this value until some number of examples have been encountered—maturation time. Note that here G&W crucially shift in footnote 28 from the criterion of convergence with probability 1 (Gold's (1967) criterion) to convergence with high probability, $1 - \delta$, where δ can be made arbitrarily small by making maturation time arbitrarily large. We show how to compute δ precisely as a function of maturation time.
- We provide simple necessary and sufficient conditions for learnability, and a natural distinction between paths and links: we show that a "local" trigger is simply an ingoing link on the Markov chain (see figure 1). Local maxima are the same as the absorbing states of the Markov chain and so we can simply use existing mathematical theory to calculate these without error.
- We show that if the learner drops either or both of the Greediness and Single Value Constraints, the resulting algorithm not only avoids the local maxima problem entirely but also converges faster (in the sense of requiring fewer examples) than the TLA. In light of this result, the question arises how strongly one should stick to these constraints, since some of these algorithms do not appear to violate any of the conservatism, cognitive load, and naturalness criteria that G&W advance as arguments for the Greediness and Single Value Constraints.

2 Refining Gibson and Wexler's Analysis: A Complete Set of Problematic Initial Grammars

G&W's TLA (and the algorithm in their appendix A) divides the set of all grammars into two disjoint sets: those that are *connected* to a chosen target grammar (via some chain of triggers) and those that are *unconnected* to the target grammar (by any chain of triggers). "Given the matrix of connected grammars, the local maxima fall out as simply the grammars that are not connected to their respective target grammars" (G&W 1994:452).

If we explicitly draw out the topology implied by this algorithm, then for the case in which grammar 5 ("English"; SVO −V2) is the target, we arrive at the picture in figure 1. This picture tells *almost* all one needs to know about the learnability of the grammar space, and is of course identical to the connectedness calculation given by G&W in their appendix A. (Also see table 2 in appendix B below, which gives the correspondence between surface unembedded phrase sequences like SVO−V2 and G&W's binary parameter triples; this information is taken directly from G&W 1994 (see their table 3).)

However, G&W's calculation does not correctly establish the list of initial states from which the learner will not converge to the target with probability 1. If there exists some path from an

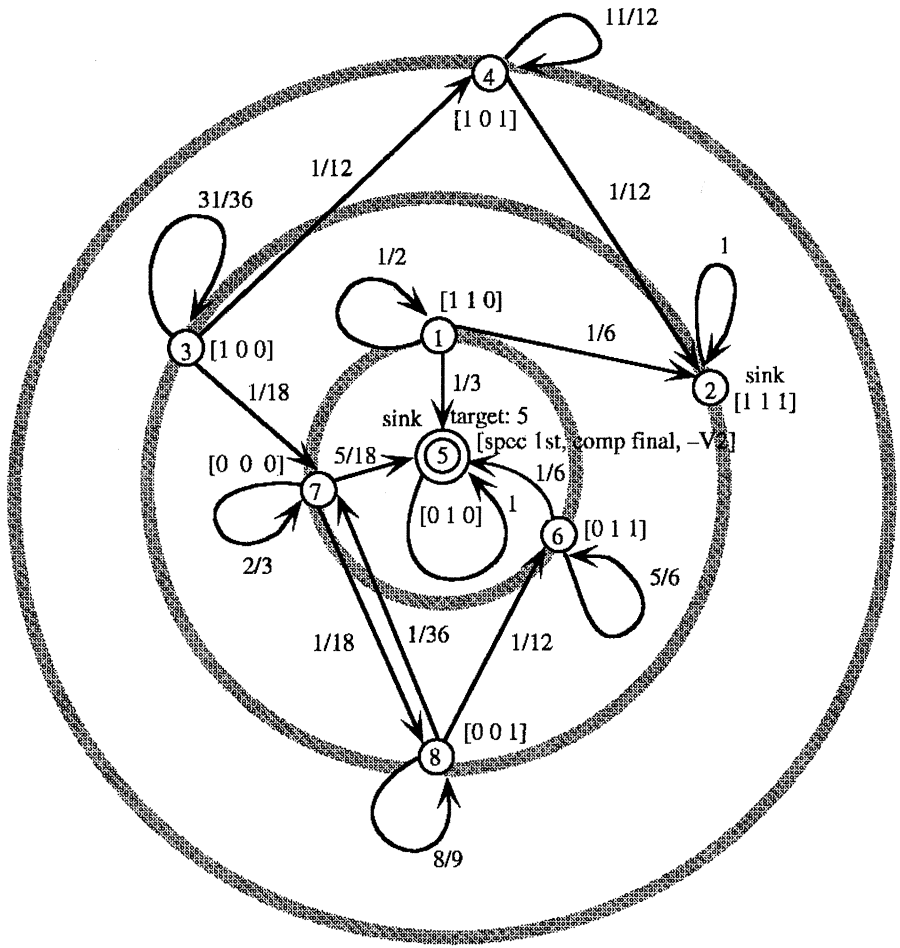

Figure 1

The eight parameter settings in G&W's example, shown as a Markov structure. Directed arrows between circles (states, parameter settings, grammars) represent possible nonzero (possible learner) transitions. The target grammar (in this case, number 5 (see table 2 of this article and table 3 of G&W 1994), setting [0 1 0]) lies at dead center. Surrounding it are the three settings that differ from the target by exactly one binary digit; surrounding those are the three settings that differ from the target by two binary digits; and the third ring out contains the single setting that differs from the target by three binary digits. Note that the learner can either cycle or step in or out one ring (binary digit) at a time, according to the single-step learning hypothesis; but some transitions are not possible because there is no input item to drive the learner from one state to the other under the TLA. Numbers on the arcs denote transition probabilities between grammar states; these values are not computed by G&W's algorithm.

initial state to a nontarget state, then there is always some finite probability that the learner will take this faulty path; thus, there is always some finite probability that the learner will not converge to the target (with probability 1). In the face of particularly "malicious" input distributions, this probability could in fact be very high, nearly 1, as we will show. Evidently, G&W assume that *if* a (triggered) path exists to the target, then it *will* be taken; however, this presumption is incorrect.

Put another way, G&W do not compute transition probabilities between the grammars (states), and by their own assumptions, the determination of learnability crucially depends on whether the learner reaches a target grammar with probability 1 (the usual Gold-type assumption; see G&W 1994:433 fn. 28). It is therefore essential to calculate these transition probabilities, which are based on the transition probabilities from one grammar to another, given some target grammar and set of positive example sentences—for instance, the 12 "degree-0" strings such as *S V, S V O,* and *S Aux V O* that are possible when the target grammar is grammar 5 ("English") in G&W's three-parameter example. In figure 1 we have labeled the grammar-to-grammar connections with these probabilities; this turns out to be a straightforward calculation. Roughly, the probability of moving from one grammar G_i to another G_j is just a measure of the number of target grammar sentences—triggers— that are in G_j but not in G_i, normalized by the total number of positive examples and the alternative grammars the learner can move to. In the case of finite sets, as in the three-parameter example G&W examine, this is a particularly simple calculation; for instance, since there are precisely 2 target strings, *S V* and *Adv S V,* that grammar 7 has but grammar 3 does not (grammar 3 does not contain any target grammar sentences; that is, $L_3 \cap L_5 = \emptyset$), the probability of moving from grammar 3 to grammar 7 is $2/12 * 1/3 = 1/18$ (there are 3 alternative grammars that are 1 bit away from grammar 3). Note how this calculation explicitly demonstrates that the whole notion of triggering is a purely extensional one, in the sense that the calculation is based solely on the *languages* generated by the space of grammars; the transitions do not connect in any other logical way to the grammars themselves.

Let us now consider figure 1 in more detail and describe how it characterizes learnability in G&W's three-parameter space. Given a target grammar at the center, an unlearnable initial grammar is one from which the learner will not converge to the target (with probability 1). G&W correctly show that states 2 and 4 are unlearnable in this sense. However, now consider state 3. Note that there is a path from this state to the target grammar 5. G&W assume that because this is the case, state 3 is not a problematic initial state. But in fact there is also a path that connects state 3 to grammar 2, a nontarget grammar. In other words, there is a positive (finite) probability that a learner starting in state 3 will converge to nontarget state 2, just as there is a finite probability that the learner will converge to target state 5. In fact, using the transition probabilities, we can calculate that a learner starting in state 3 converges to target grammar 5 with probability exactly 0.6, not probability 1 as G&W require, and so converges to the nontarget grammar with probability 0.4. Note that this misconvergence probability is not insignificant. In sum, state 3 is not learnable under G&W's (Gold-type) assumption of convergence in the limit to the correct target grammar with probability 1. Note finally that state 3 is also a $-$V2 grammar, so that the set of problem states is not confined to those that are $+$V2, contrary to the situation described by G&W.

Using this picture, we can also now readily interpret some of G&W's terminological notions. A *local trigger* is simply a datum that would allow the learner to move along an *ingoing* link in the figure. For example, the link from grammar state 3 to grammar state 7 does correspond to a local trigger, as does the link from grammar state 4 to grammar state 2; however, the link from grammar state 3 to grammar state 4 is not a local trigger. Also, because of the Single Value and Greediness Constraints, the learner can only either (a) stay in its current state; (b) move one step inward (a local trigger); or (c) move one step outward (note that this also happens given data from the target, just as in case (b)). These are the only allowed moves; the learner cannot move to another state within the same ring.

The learnability properties of this space can also be described more formally once it is recognized (as G&W note (1994:412 fn. 11)) that the parameter space forms a Markov chain: that is, a finite set of states with appropriate transition probabilities (see Isaacson and Madsen 1976 for a formal definition). In this Markov chain, certain states have no outgoing arcs; these are among the *absorbing states* because once the system has made a transition into one of these states, it can never exit. More generally, let us define the set of *closed states* to be any proper subset of states in the Markov chain such that there is no arc from any of the closed states to any other state in the Markov chain.

Note that in the systems under discussion the target state is always an absorbing state (once the learner is at the target grammar, it can never exit), so the Markov chains we will consider always have at least one absorbing state. In the example three-parameter system, state 2 is also an absorbing state. Given this formulation, a very simple and now corrected criterion for the learnable initial states (with respect to some target grammar) can immediately be formulated.

Theorem 1 *Given a Markov chain C corresponding to a parameter space, a target parameter setting, and a TLA learner that attempts to learn the target parameters, there is exactly one absorbing state (corresponding to the target grammar) and no other closed state (distinct from and not including the target state) iff target parameters can be correctly set by the TLA in the limit (with probability 1).*

Proof Sketch $\Leftarrow$. By assumption, C is learnable. Now assume for sake of contradiction that there is more than one closed state. Pick the closed state that is not the target state. If the learner starts in this state, it can never reach the target absorbing state, by the definition of a closed state. This contradicts the assumption that the space was learnable.

$\Rightarrow$. Assume that there exists exactly one absorbing state in the Markov chain M and no other closed state. There are two cases. Case (i): At some time the learner reaches the target state. Then, by definition, the learner has converged and the system is learnable. Case (ii): There is no time at which the learner reaches the target state. Then the learner must move among a set of nontarget states. But this by definition forms a closed set of states distinct from the target, a contradiction.[3] ■

[3] This argument can be made more precise by using the standard decomposition of a finite Markov chain into its transient states and closed equivalence classes of recurrent states, and then showing that all nontarget states are transient ones. This implies that the learner will be at a nontarget state with probability 0 in the limit and at the target state with probability 1. Such a formal argument is developed in Niyogi and Berwick, forthcoming.

Corollary 1 *Given a Markov chain C corresponding to a TLA learner, the set of learnable initial states is exactly the set of states that are connected to the target and unconnected to the nontarget closed states of the Markov chain.*

We are now in a position to state and correct the flaw in the algorithm, step 3 of G&W's appendix A, that computes problem states.

3. For each target grammar G_{target}:
   For each source grammar G_{source}:
   If G_{target} is not in CONNECTED-GRAMMARS (G_{target}, G_{source})
   Then add the pair (G_{source}, G_{target}) to LOCAL-MAXES.
4. Return LOCAL-MAXES. (G&W 1994:450)

This algorithm computes as unlearnable initial grammars those that are unconnected to the target grammar (implicitly, the learnable grammars are those that are connected to the target). But as we have just established, this is false. For example, consider state 4 in figure 1: this is unconnected to the target and so by G&W's algorithm it is unlearnable. Their conclusion is true; however, the reason for nonlearnability is not. State 4 is not learnable because it is connected to the nontarget absorbing state 2.

Assuming Markov chains whose only closed states are absorbing states, as is the case in the three-parameter system, we can present a corrected version of G&W's algorithm for finding the complete set of problem states. A more complex revision would be required to handle Markov chains with closed states of other kinds (essentially, cycles).

Get set LOCAL-MAXES returned by G&W's algorithm;
For each G_{target},
For each G_L such that $(G_L, G_{target}) \in$ LOCAL-MAXES:
For each G_{source},
If $G_L \in$ CONNECTED-GRAMMARS (G_{target}, G_{source})
Then add (G_{source}, G_{target}) to LOCAL-MAXES.

This clearly carries out a different computation for LOCAL-MAXES, as noted above and shown in figure 1. Now we can compute a *complete* list of (initial state, target state) pairs such that the learner will not converge to that target grammar from that initial grammar (with probability 1); this list is shown in table 1.

3 Consequences of the Revised Account

3.1 The Single Value and Greediness Assumptions

The remainder of G&W 1994 turns on the analysis of the local maxima problem, which is basically a consequence of G&W's adoption of the Single Value Constraint and the Greediness Constraint. It is therefore also crucial to examine the grounds for these assumptions, as G&W note. We consider Greediness first, and then the Single Value Constraint.

Table 1

Complete list of problem states, that is, all combinations of starting grammar and target grammar that result in nonlearnability of the target. The items marked with an asterisk are those pairs not listed in G&W 1994.

Initial grammar	*Target grammar*	*State of initial grammar (in Markov structure)*	*Probability of not converging to target*
(SVO − V2)*	(OVS − V2)		0.5
(SVO + V2)	(OVS − V2)	Absorbing state	1.0
(SOV − V2)*	(OVS − V2)		0.15
(SOV + V2)	(OVS − V2)	Absorbing state	1.0
(VOS − V2)*	(SVO − V2)		0.33
(VOS + V2)	(SVO − V2)	Absorbing state	1.0
(OVS − V2)*	(SVO − V2)		0.33
(OVS + V2)	(SVO − V2)		1.0
(VOS − V2)*	(SOV − V2)		0.40
(VOS + V2)	(SOV − V2)	Absorbing state	1.0
(OVS − V2)*	(SOV − V2)		0.08
(OVS + V2)	(SOV − V2)	Absorbing state	1.0

3.1.1 Greediness Constraint Recall that the TLA is a greedy algorithm; that is, the learner will make a change in its parameter settings only if the new parameter setting allows it to analyze the input sentence whereas the current one does not. Does such a greediness assumption matter? G&W's arguments for the Greediness Constraint boil down to three: (a) conservatism—one should prefer small changes in the currently hypothesized grammar[4] to larger changes; (b) cognitive load; and (c) linguistic naturalness. Let us consider each of these in turn, putting aside the Single Value Constraint for the moment to isolate the effects of Greediness.

(a) *Conservatism.* Does Greediness entail conservatism? Note first that Greediness is not a "batch" constraint; that is, it applies, not to a *set* of sentences, but to a *single* sentence, the current input sentence. We must further distinguish between two kinds of conservatism: *intensional* conservatism, that is, small changes in parameter space or grammar space; and *extensional* conservatism, that is, small changes in the language the learner can analyze (roughly, external "linguistic behavior"). Finally, note that unless some assumption is made about a "smoothness" relation between grammars and languages, these two notions remain distinct (small changes in grammar space need not translate into small changes in language space, and vice versa).[5]

[4] Or, possibly, the currently hypothesized language.

[5] In fact, it is not at all obvious what the relation between grammar/parameter space and language space is, a matter of some importance for any sentence-driven learning algorithm. As far as we know, this is a mathematically difficult

When we examine this point, we find that Greediness (without the Single Value Constraint) apparently allows the learner to make massive jumps through either grammar space or language space, contrary to what G&W desire. In the case of grammar space, the learner can make many parameter changes just to account for one sentence, the current datum. So Greediness by itself does not meet the test of grammar conservatism. This is to be expected. Depending upon the learner's current hypothesis, and depending upon the structure of the parameter space, Greediness could move the learner far away from its current hypothesis. In fact, a passage in which G&W justify the Single Value Constraint argues precisely the same point.

> The need to account for an unanalyzable input item can lead the learner to shift to a grammar that is nothing like the current one, thus allowing the possibility of a massive immediate change in her linguistic behavior. (G&W 1994:442)

For example, in the three-parameter system, suppose the learner was in state 1 (Spec-final, Comp-final, −V2). A sentence of the form *S V* or *S Aux V* is in grammar state 4 but not grammar state 1. Therefore, the requirement to analyze this string could force the learner to state 4, the grammar *farthest* from the target and grammar 1. This is precisely the behavior that G&W wanted to avoid, and it arises precisely because Greediness holds. In short, we see no logical relation between Greediness and grammatical conservatism; rather, this conservatism is imposed by the Single Value Constraint.

Turning to extensional conservatism, here too we find no necessary relation between Greediness and small changes in language space (the set of sentences the learner can analyze at any one point). It is conceivable (as we show by example below) that there exist two grammars, G_1 and G_2, such that extensionally G_1 is closer to the target than G_2, but Greediness forces the learner from G_1 to G_2.[6] It is easy to see why: any sentences in G_2 (and the target) presented to the learner would, by Greediness, drive the learner *farther* from the target, to G_2. Whether or not this happens is not a property of Greediness (local hill-climbing); rather, it is a property of the language space. We cannot know whether Greediness on the average drives the learner closer to or farther away from the target—how far the learner moves and how fast—without actually calculating transition probabilities with respect to a specific space; note that these transition probabilities are dependent on extensional (language) set intersections.

An example from the three-parameter system should make this point clear. Turning again to figure 1, note that grammars 6 and 7 are equidistant from the target in grammar space; they each differ from the target by one binary digit (one parameter setting). However, although grammar 6 is quite close to the target in language space (it has 6 degree-0 strings in common with the

---

question that has not really ever been addressed, although perhaps some notion of approximation by power series, as proposed in works by, for example, Chomsky and Schutzenberger (1963), could prove relevant here. We are currently exploring this question since there is an abundant mathematical literature on functional approximation in such spaces.

[6] This presumes that it is possible to put a measure on the extensional properties of the "surface strings" like *S V O*. This is clearly possible in the case of finite sets; it is also possible for infinite sets if certain measurability conditions are met. A discussion of these technical details would lead us too far afield at this point.

target, out of 12), grammar 7 is farther away in language space (it has only 2 degree-0 strings in common with the target: *S V, Adv S V*). (This example also demonstrates our point that one cannot assume any kind of necessarily smooth relation between grammar space and language space.)

Now, again crucially assuming no Single Value Constraint, if the learner is in state 6 and is then presented with the example *Adv S V,* by Greediness it will be forced to move to state 7—farther away from the target in language space.

In sum, Greediness has nothing to do with either grammar or language conservatism.

(b) *Resource limitations.* Does Greediness help the learner in terms of cognitive or computational resources, that is, time or space (or variants thereof, such as memory load)? Implicitly, one of G&W's ideas here seems to be that without Greediness the learner might wander all over the parameter space before converging, hence taking more time.

> Without the Greediness Constraint, the learner can radically alter her grammatical hypothesis very quickly, no matter how close to the target she has come. For example, suppose that the learner has set all but one of her parameters correctly, and that she is presented with a sentence pattern from the target grammar that is not in the current grammar. A nongreedy learner might change any of the parameters because of the unanalyzable data. . . . Lacking the Greediness Constraint makes learning the grammar very difficult: a learner might be next to the target many times over, each time with a different parameter set incorrectly, before she finally accidentally achieves the target. (G&W 1994: 443)

However, is this true in fact? Intuitively, although Greediness allows the learner to "spiral away" from the target, it might also allow the learner to "spiral in." Once again, the exact balance between the two cannot be determined via intuition. One must either carry out a formal proof or do sample calculations. For calculations, once again we need the actual transition probabilities. In fact, when we carry out simulations on the three-parameter space, we discover that a nongreedy learner is indeed faster than G&W's greedy learner, as shown in figure 2.

Further, not only does the greedy algorithm take more time, there is also a sense in which it requires more computation at any single step than a nongreedy one. Suppose the learner has received a sentence and is not able to analyze it in its current state. Greediness requires determining whether the new grammar can allow the learner to analyze the input or not. Nongreediness does not need to carry out this test; the learner can simply move to another state.

G&W also aim to keep memory load low by using an incremental algorithm that processes only one sentence at a time, rather than a batch algorithm that stores up many sentences and processes them as a group. Behind this assumption lies one about cognitive/memory load: the learner should not have to keep many sentences in memory, which seems reasonable enough.

However, this point is orthogonal to the question of Greediness, because all the variants that we consider in figure 2 and that do better than the greedy algorithm are also incremental. For instance, take the simplest possible algorithm variant, which we will call *Random Step* (that is, an algorithm with no Greediness Constraint but with the Single Value Constraint). With Random Step, if the learner is in some grammar state and receives a sentence it cannot analyze, it simply picks a grammar state at random to move to (no more than one parameter setting away from the

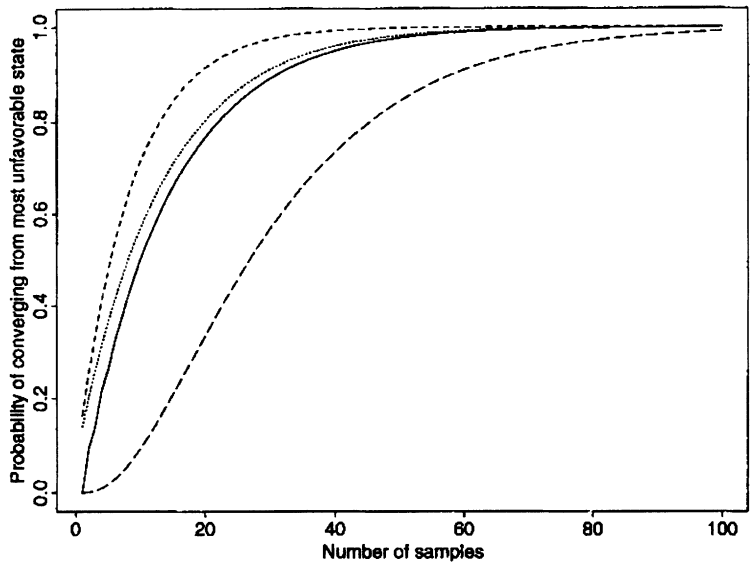

Figure 2

Convergence rates for different learning algorithms when L_1 is the target language. The curve with the slowest rate (large dashes) represents the TLA. The curve with the fastest rate (small dashes) represents the algorithm Random Step with no Greediness Constraint or Single Value Constraint. Rates for Random Step with either the Greediness Constraint or the Single Value Constraint lie between these two and are very close to each other. The curves for cases where the target is some language other than L_1 are similar, though not depicted here; Random Step generally dominates the TLA.

current grammar). This algorithm also processes only one sentence at a time; further, as noted above, its cognitive load is actually less than that of the TLA. Yet Random Step works faster than the TLA in the three-parameter space, and it does not incur any local maxima problems. Other incremental learning algorithm variants also work faster.

In short, we do not see any substantive argument for Greediness on computational/cognitive resource grounds either. Our simulations show that not only do nongreedy learners learn faster, contrary to G&W's intuitions, they also never get stuck. They can also be conservative. Nongreediness would therefore seem to have all the advantages of Greediness and none of its disadvantages, at least with respect to learning time.

(c) *Linguistic naturalness* is thus the remaining possible argument for Greediness. We have been unable to think of a good definition or concrete examples for linguistic naturalness.

3.1.2. Single Value Constraint G&W also advance arguments for changing only one parameter value at a time: again, conservatism, cognitive load, and naturalness. Here we agree that the Single Value Constraint does support the properties of conservative hypothesis formation (at least in grammar space, not necessarily in language space) and cognitive load.

It is clear that the Single Value Constraint enforces grammatical conservatism. The learner moves at most one step in parameter (grammatical hypothesis) space. However, bearing in mind our earlier discussion on the "smoothness" properties of grammar parameter space, it is difficult to predict whether any kind of language conservatism follows as a result.

When we consider cognitive load, it is not immediately clear whether dropping the Single Value Constraint would dramatically increase the resources the learner has to use. For example, G&W argue that

> [t]he Single Value Constraint, coupled with the constraint that allows only one new grammatical hypothesis to be formed and tested per input item, ensures that only limited resources are necessary at each step in the parameter-setting process. (G&W 1994:442)

The constraint that only one new grammatical hypothesis be formed and tested per input item is distinct (as G&W themselves admit) from the Single Value Constraint. It is this incremental data constraint that lightens the cognitive burden on the learner and could just as well be maintained while dropping the Single Value Constraint. For instance, doing Random Step incrementally—moving to only one new grammar state at a time—imposes the same computational burden.[7]

3.2 Sample Complexity or the Poverty of Stimulus Revisited

Our Markov analysis goes beyond clarifying the asymptotic (in the limit) learnability properties of the TLA and parameter spaces. It also sheds valuable light on the sample complexity of the language acquisition problem. Specifically, it allows us to actually count the number of sentences the child will have to hear before converging to the target with high probability—that is, it actually measures the number of stimuli needed in the "poverty of stimulus" sense. Recall that the TLA learner moves from state to state according to the transition probabilities described earlier. Suppose the learner starts out in some state i of the chain. After receiving m examples, the learner might be in one of a number of states. Using Markov chain theory, it is possible to exactly compute $p_i(m)$, the probability that the learner is in the target grammar after m examples. Clearly, learnability requires that

$$\forall i, \lim_{m \to \infty} p_i(m) = 1,$$

in other words, with probability 1, that the learner will converge to the target. Figure 2 shows a plot of $p_i(m)$ for the most unfavorable starting state (most unfavorable i) as a function of m. In this case, the target language is L_1 and there are no local maxima; in other words, the learner converges from every starting state. Here we see that the probability that the learner has arrived

[7] One might finally argue that changing more than one parameter at a time is computationally more complex than changing a single parameter; however, flipping just two parameter switches at a time instead of one imposes a minimal additional burden. It is of course stronger to assume one possible change instead of two, and one is the natural stopping point if there is no other known limit, but the question needs more examination. There could be some trade-off between convergence time and number of data examples that can be examined at one step.

at the target after 100 examples is almost 1. This analysis thus allows us to quantify the sample complexity of the problem, precisely the poverty of the stimulus.

The sample complexity of language acquisition is of crucial importance; it has not often been explicitly addressed.[8] Most research tends to concentrate on learnability in the limit. However, an adequate computational model of language acquisition must not only converge to the target in the limit but also do so in reasonable time with a suitably small number of examples. After all, the poverty of stimulus that the human child experiences during language acquisition is what motivated the development of more constrained models of linguistic structure and is at the heart of the rationalist approach to linguistics. Given the mathematical formulation presented here, average and standard deviations for convergence times can be computed with respect to different sample distributions, that is, different distributions of positive example inputs the learner would receive. As far as we know, this is the first such analysis that has been advanced. Critical questions remain, of course: for one thing, one would like to know what the sample complexity is for a "real" parameter space, in other words, what happens to the convergence time as the number of parameters grows. If the TLA (or Random Step) requires an exponential number of positive examples given actual input distributions, that could to our minds violate a cognitive fidelity criterion, similar to the one about degree-0 sentences or cognitive load.[9]

4 A Note on Gibson and Wexler's Maturational Solution

As a consequence of our precise Markov model, we are able to compute convergence times in a variety of situations. We have already remarked on how to do this in the general case when there are no default parameters (that is, all parameters are open to change by the learner). This yields the Markov structure of figure 1. G&W propose a maturational solution to the local maxima problem. In particular, they suggest that the V2 parameter be set to a default value of $-$V2 and that the learner not be allowed to change it until some period of (maturational) time has elapsed; after this time, all parameters are available to the learner.

With respect to our Markov structure this suggestion has two effects:

1. It restricts the learner to start in only the $-$V2 states of the chain; in figure 1 this corresponds to restricting starting conditions to states 1, 3, 5, and 7.
2. All links from $-$V2 to $+$V2 are eliminated. Thus, the link from state 3 to state 4 is eliminated, as is the link from state 7 to state 8.

The topology and corresponding transition probabilities of the constrained Markov chain can now be recomputed as before. Since the learner never reaches local maxima before maturation time has elapsed, the problem of getting stuck in an absorbing state never arises. However, some of the $-$V2 states are still problematic; as discussed earlier, the learner could be in one of those

[8] But see Osherson, Stob, and Weinstein 1986:chap. 8 for one general formalization, dubbed "text efficiency," that allows for the existence of abstract sample complexity hierarchies. However, the actual sample complexity bounds are not in general explicitly constructed.

[9] Using the Markov model, one should also be able to develop a mathematical analysis for the average number of "grammar changes" the learner makes before arriving at the target grammar—a useful measure of conservatism. Such results are typically available in Markov theory, so it should be possible to calculate something similar here; we are currently developing this approach. We thank Janet Fodor for this suggestion.

problematic states even after maturation. In this case the learner would not converge to the target with probability 1. Thus, if a finite maturation time is allowed (say, time t), then the learner will not converge to the target with probability 1, but instead will converge only with probability $1 - \delta$, where δ (the residual probability of misconvergence) depends upon the time t and goes to 0 as t goes to infinity. This fact is noted by G&W as a "crucial assumption" (1994:433 fn. 28) and in fact marks a shift, as they say, from Gold's definition of identification in the limit with probability 1 to identification with probability less than 1. That is, the maturational solution G&W propose works by dropping their initial assumption of identification in the limit.

Once Gold's assumption is dropped, to judge the psychological plausibility of the maturational solution it becomes necessary to quantify the relation between maturation time t and δ. This again demands some calculations.

As a concrete example of the power of our formulation, we can carry out a simple calculation to show what the relation between maturational time t and δ is like. Imagine that the learner starts out in state 3, a permissible $-$V2 starting state. As a result of maturational constraints, the link from state 3 to state 4 is eliminated and only two loops from state 3 are left in the revised Markov chain. It can be shown that the learner will, with probability 31/36, remain in grammar state 3 after a single input and will, with probability 5/36, move to state 7. The probability that after t inputs (corresponding to maturation time) the learner still remains in state 3 is simply $(31/36)^t$. Now the V2 parameter becomes active, the complete Markov chain of figure 1 reigns, and the learner will with probability 2/5 (by our previously shown computations) converge to state 2 and thus never reach the target. Thus, the total probability of not converging to the target in the maturational case is $\frac{2}{5}(\frac{31}{36})^t$. This will be greater than δ if t is less than $1/\log(36/31) \log(2/5\delta)$. Thus, t is $O(\ln(1/\delta))$. Computations like this can be carried out for each state precisely because of our Markov formulation.[10]

5 Conclusion

In this article we have tried to show that a complete analysis of learnability in parameter spaces hinges on a more precise probabilistic analysis than that presented by G&W. The reanalysis is revealing. Simple incremental algorithms like Random Step converge faster than the TLA, can be conservative, and yet do not suffer from G&W's local maxima problem. To this extent, the whole problem of local maxima taken up by G&W is more apparent than real, and G&W's proposed solutions are unneeded.[11]

[10] Why does G&W's maturational solution work? G&W seem to say that it works because all the local maxima are +V2.

> Note that each of the local maxima has the V2 parameter set to +V2 and that the target grammar in each case has the V2 parameter set to −V2. As a result of this pattern, it turns out that local maxima can be avoided altogether if the V2 parameter has the default value −V2, and the values for the other two parameters are initially unset. (G&W 1994:430)

However, as we have just shown, this presumption is false (see also table 1). Some local maxima have the V2 parameter set to −V2. Fortunately, it is the case that all the *strictly* absorbing states are +V2. This is the real reason why the default setting/maturational argument works.

[11] It should be pointed out that Random Step's superiority in this case might be due entirely to the choice of parameters, that is, to the shape of the hypothesis space. For some other parametric space, Random Step might not converge faster; however, it will always avoid the local maxima problem.

Despite these differences that ensue, we would like to point out that many of G&W's arguments are correct in spirit. For example, the local maxima issue is indeed central to the learnability question and triggers in a parametric framework. However, the reasoning needs to be complete. If anything, our own attempts to formulate a correct learnability convergence proof show how subtle a seemingly intuitive idea like convergence in a finite space can be. Intuitions must be backed by exact formulations.

Perhaps the most important other difference that emerges from our reanalysis of G&W 1994 is that as far as we can tell, the Greediness assumption does not seem to have foundation. This also renders the local maxima problem moot.

Appendix A: Derivation of the Transition Probabilities

We have argued that the TLA working on finite parameter spaces reduces to a Markov chain. This argument cannot be complete without a precise computation of the transition probabilities from state to state. We do this now.

Consider a parametric family with n Boolean valued parameters. These define 2^n grammars (and by extension, languages), as we have discussed. Let the target language L_t consist of the strings (sentences) $s_1, s_2, \ldots$, that is,

$$L_t = \{s_1, s_2, s_3, \ldots\} \subseteq \Sigma^*.$$

Let there be a probability distribution P on these strings,[12] according to which they are drawn and presented to the learner. Suppose the learner is in a state s corresponding to the language L_s. Consider some other state k corresponding to the language L_k. What is the probability that the TLA will update its hypothesis from L_s to L_k after receiving the next example sentence? First, observe that as a result of the Single Value Constraint, if k and s differ by more than one parameter setting, then the probability of this transition is 0. As a matter of fact, the TLA will move from s to k only if the following two conditions are met: (a) the next sentence it receives (say, ω, which occurs with probability $P(\omega)$) is analyzable by the parameter settings corresponding to k and not by the parameter settings corresponding to s, and (b) upon being unable to analyze ω, the TLA has a choice of n parameters to change, and it picks the one that would move it to state k.

Event (a) occurs with probability $\sum_{\omega \in L_k \setminus L_s} P(\omega)$ whereas event (b) occurs with probability $1/n$ since the parameter to change is chosen uniformly at random out of the n possible choices. Thus, the cooccurrence of both of these events yields the following expression for the total probability of transition from s to k after one step:

$$P[s \rightarrow k] = \sum_{s_j \notin L_s, s_j \in L_k} (1/n) P(s_j).$$

[12] This is equivalent to assuming a noise-free situation, in the sense that no sentence outside of the target language can occur. However, one could choose malicious distributions so that all strings from the target were not presented to the learner. If one wished to include noise, one would only need to consider a distribution P on Σ^* rather than on the strings of L_t. Everything else in the derivation would remain identical. This would yield a Markov chain corresponding to the TLA operating in the presence of noise.

Since the total probability over all the arcs out of s (including the self loop) must be 1, the probability of remaining in state s after one step is

$$P[s \rightarrow s] = 1 - \sum_{k \text{ is a neighboring state of } s} P[s \rightarrow k].$$

Finally, given any parameter space with n parameters, we have 2^n languages. Fixing one of them as the target language L_t, we obtain the following procedure for constructing the corresponding Markov chain. Note that this will yield a Markov chain with the same topology (in the absence of noise) as the procedure that G&W propose. However, our procedure differs significantly from G&W's in adding a probability measure on the language family.

- (Assign distribution) First fix a probability measure P on the strings of the target language L_t.
- (Enumerate states) Assign a state to each language (i.e., each L_i).
- (Take set differences) Now, for any two states i and k, if they are more than 1 Hamming distance apart, then the transition $P[i \rightarrow k] = 0$. If they are 1 Hamming distance apart, then $P[i \rightarrow k] = \frac{1}{n}P(L_k \setminus L_i)$.

Appendix B: Example Calculation

For the three-parameter system studied by G&W, the transition probabilities can be computed straightforwardly according to the procedure outlined above. Consider, for example, the transition from grammar state 3 (OVS −V2) to grammar state 7 (SOV −V2) when the target is state 5 (SVO −V2). This is shown in figure 1. According to G&W's data reproduced in table 2, there are 12 sentences in the target that the learner is likely to receive. If these sentences occur with equal likelihood, then the learner will move to grammar state 7 if and only if the following two events occur:

Event 1

A sentence occurs that can be analyzed by grammar state 7 but not by grammar state 3. There are two such sentences; consequently, the probability of this event is $\frac{2}{12}$.

Event 2

Given the occurrence of event 1, the learner will attempt to change one of the parameters at random. Of the three parameters it can change, only the "Spec" parameter will move it to grammar state 7. This occurs with probability $\frac{1}{3}$ if the learner changes the parameter uniformly at random.

Hence, the transition from state 3 to state 7 occurs when both events 1 and 2 occur. The total probability of this is $\frac{2}{12} \cdot \frac{1}{3} = \frac{1}{18}$. This is how the transition probability of figure 1 is obtained. Other transition probabilities are obtained in similar fashion.

Appendix C: Unembedded Sentences for Parametric Grammars

Table 2 provides the unembedded (degree-0) sentences from each of the eight grammars (languages) obtained by setting the three parameters to different values (see G&W 1994:tab. 3). The languages are referred to as L_1 through L_8.

Table 2

The eight grammars defined by G&W, along with their associated parameter settings and unembedded surface strings. (See G&W 1994:tab. 3.)

Language	*Spec*	*Comp*	*V2*	*Degree-0 unembedded sentences*
L_1 VOS − V2	1	1	0	V S, V O S, V O1 O2 S Aux V S, Aux V O S, Aux V O1 O2 S, Adv V S Adv V O S, Adv V O1 O2 S, Adv Aux V S Adv Aux V O S, Adv Aux V O1 O2 S
L_2 VOS + V2	1	1	1	S V, S V O, O V S, S V O1 O2, O1 V O2 S, O2 V O1 S S Aux V, S Aux V O, O Aux V S S Aux V O1 O2, O1 Aux V O2 S, O2 Aux V O1 S Adv V S, Adv V O S, Adv V O1 O2 S Adv Aux V S, Adv Aux V O S, Adv Aux V O1 O2 S
L_3 OVS − V2	1	0	0	V S, O V S, O2 O1 V S V Aux S, O V Aux S, O2 O1 V Aux S, Adv V S Adv O V S, Adv O2 O1 V S, Adv V Aux S Adv O V Aux S, Adv O2 O1 V Aux S
L_4 OVS + V2	1	0	1	S V, O V S, S V O, S V O2 O1, O1 V O2 S, O2 V O1 S S Aux V, S Aux O V, O Aux V S S Aux O2 O1 V, O1 Aux O2 V S, O2 Aux O1 V S Adv V S, Adv V O S, Adv V O2 O1 S Adv Aux V S, Adv Aux O V S, Adv Aux O2 O1 V S
L_5 SVO − V2	0	1	0	S V, S V O, S V O1 O2 S Aux V, S Aux V O, S Aux V O1 O2, Adv S V Adv S V O, Adv S V O1 O2, Adv S Aux V Adv S Aux V O, Adv S Aux V O1 O2
L_6 SVO + V2	0	1	1	S V, S V O, O V S, S V O1 O2, O1 V S O2, O2 V S O1 S Aux V, S Aux V O, O Aux S V S Aux V O1 O2, O1 Aux S V O2, O2 Aux S V O1, Adv V S Adv V S O, Adv V S O1 O2, Adv Aux S V Adv Aux S V O, Adv Aux S V O1 O2
L_7 SOV − V2	0	0	0	S V, S O V, S O2 O1 V S V Aux, S O V Aux, S O2 O1 V Aux, Adv S V Adv S O V, Adv S O2 O1 V, Adv S V Aux Adv S O V Aux, Adv S O2 O1 V Aux
L_8 SOV + V2	0	0	1	S V, S V O, O V S, S V O2 O1, O1 V S O2, O2 V S O1 S Aux V, S Aux O V, O Aux S V S Aux O2 O1 V, O1 Aux S O2 V, O2 Aux S O1 V Adv V S, Adv V S O, Adv V S O2 O1 Adv Aux S V, Adv Aux S O V, Adv Aux S O2 O1 V

References

Chomsky, Noam, and Marcel P. Schutzenberger. 1963. The algebraic theory of context-free languages. In *Computer programming and formal systems*, ed. P. Braffort and D. Hirschberg, 118–161. Amsterdam: North-Holland.

Clark, Robin, and Ian Roberts. 1993. A computational model of language learnability and language change. *Linguistic Inquiry* 24:299–345.

Gibson, Edward, and Kenneth Wexler. 1994. Triggers. *Linguistic Inquiry* 25:407–454.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Isaacson, David, and John Madsen. 1976. *Markov chains*. New York: Wiley.

Niyogi, Partha, and Robert Berwick. 1993. Formalizing triggers: A learning model for finite spaces. AI memo 1449, CBCL Memo 86. MIT, Cambridge, Mass.

Niyogi, Partha, and Robert Berwick. Forthcoming. A language learning model for finite parameter spaces. *Cognition*.

Osherson, Daniel, Michael Stob, and Scott Weinstein. 1986. *Systems that learn*. Cambridge, Mass.: MIT Press.

Wexler, Kenneth, and Peter Culicover. 1980. *Formal principles of language acquisition*. Cambridge, Mass.: MIT Press.

Center for Biological and Computational Learning
E25-201
45 Carleton Street
MIT
Cambridge, Massachusetts 02139

berwick@ai.mit.edu
pn@ai.mit.edu