Spatio-Temporal Evolution of Grammars

Sumeet Agarwal

February 1, 2008

1 Introduction

Grammar can be thought of as the computational system of language: it is a set of rules that specifies how to construct sentences from words. Grammar is responsible for making language such a powerful means of communication: it provides the means for mapping a finite vocabulary of words to an infinite repository of syntactic expressions. An interesting problem is how children learn grammars: it is well known that the empirical evidence available (the sentences heard from parents and others in the environment) drastically under-specifies the underlying syntactic rules. Noam Chomsky [1] and others have proposed the concept of a 'Universal Grammar' which is somehow hardwired into the brain and substantially restricts the grammar search space during the learning process.

Here, we use the mathematical model for grammar evolution proposed by Komarova *et al.* [2] (paper attached) to study how the distribution of grammars within a population changes over time. The model uses two possible kinds of learning mechanisms: *memoryless learning* and *batch learning*. This work extends the existing model in two ways: it introduces spatial variation (the original model had only temporal variation), and it uses a learning mechanism which is a linear combination of memoryless and batch learning, and is thus seemingly more biologically realistic.

2 Basic Model

The idea is to model the evolutionary dynamics of the population. Thus, each individual will have a certain 'fitness', defined by a fitness function (Eq. 1 of [2]). This measures how well the individual is able to communicate overall with others in the population (a constant background fitness is also assigned to each individual, denoted by f_0). The reproductive ability of individuals is assumed to be proportional to their fitness, and each child is expected to acquire the parent's language with a certain *learning accuracy*. These assumptions lead to the dynamics represented by the system of ordinary differential equations (2) in [2].

To make analysis easier, we simplify the system further by making it fully symmetric: i.e., we assume that the amount of overlap between any two grammars is the same (this is denoted by a, $0 \leq a \leq 1$), and also that the learning accuracy (denoted by q, $0 \leq q \leq 1$) is the same for all languages. The resulting

system (Eq. 5 of [2]) can now be analysed to find the steady states. Due to the symmetry introduced, the system is highly degenerate, which means that we need to solve it for just one language and all the others can then be written in terms of that. So, without loss of generality, we assume X to be the fraction of the population speaking language 1 (out of n languages in total), and look for fixed points of X. These are the roots of Eq. (6) of [2]. One of them is obvious: $X = \frac{1}{n}$, which corresponds to all languages being spoken with equal frequency, is bound to be a steady state given our symmetry assumptions. So we divide the given cubic equation by $(X - \frac{1}{n})$ to get the following quadratic:

$$\frac{-n}{n-1}X^2 + \left(q + \frac{1}{n-1} + \frac{1-q}{(n-1)^2}\right)X - \frac{n(1-q)}{(n-1)^2} - \frac{n(a+f_0)(1-q)}{(n-1)(1-a)} = 0 \quad (1)$$

The roots of this are given by Eq. (8) and (9) of [2]. However, the corresponding steady states only exist if the roots are real; thus the discriminant has to be non-negative. This implies that the learning accuracy, q, has to be over a certain threshold q_1, given by Eq. (10) of [2]. So for $q < q_1$, the uniform distribution is the only steady state.

2.1 Stability analysis of uniform distribution

$X_0 = \frac{1}{n}$ is the steady state; let $x = X_0 + \tilde{x}$, where $\tilde{x} << X_0$. Using $f(x)$ to denote $\frac{dx_1}{dt}$ (the right hand side of Eq. (5) of [2] for $j = 1$), we can use the Taylor expansion to write:

$$\begin{aligned} f(x) &= f(X_0 + \tilde{x}) \\ &= f(X_0) + \tilde{x}f'(X_0) + ... \end{aligned}$$

$f(X_0) = 0$ by definition, so ignoring second- and higher-order terms we get:

$$\begin{aligned} f(x) = \tilde{x}(1-a)\left[-3X_0^2 + 2X_0 q - \frac{2(1-X_0)}{n-1}\left(-X_0 + \frac{1-q}{n-1}\right) - \frac{(1-X_0)^2}{n-1}\right] \\ - \tilde{x}\frac{(1-q)(a+f_0)n}{n-1} \end{aligned}$$

Substituting $X_0 = \frac{1}{n}$ and simplifying, we get:

$$f(x) = \tilde{x}\left[\frac{-(1-a)(n+1-2qn) - (1-q)(a+f_0)n^2}{n(n-1)}\right]$$

For stability, $f(x)$ needs to have sign opposite to $\tilde{x}$. Thus, we need:

$$\begin{aligned} -(1-a)(n+1-2qn) &< (1-q)(a+f_0)n^2 \\ 2qn - n - 1 &< \frac{(1-q)(a+f_0)n^2}{1-a} \\ qn\left[2 + \frac{(a+f_0)n}{1-a}\right] &< \frac{(a+f_0)n^2}{1-a} + n + 1 \end{aligned}$$

$$\boxed{q < \frac{(a+f_0)n^2 + (n+1)(1-a)}{n[2(1-a) + (a+f_0)n]}} \quad (2)$$

Thus we see that the learning accuracy has to be below a certain threshold (the right-hand side of Equation (2), which we will denote by q_2) for the uniform distribution to be stable. Beyond that point, it becomes unstable and the only stable solution is $x_1 = X_+$ (see Eq. (8) of [2]), which is typically a number close to 1 and thus corresponds to a solution where one grammar is dominant in the population and all the others have faded away. So the model leads to the sensible result that if the learning accuracy is high enough, the population converges to a *one-grammar* solution. In the next section we will see how we can compute the learning accuracy in terms of other model parameters, on the basis of certain assumptions about how language learning occurs.

3 Learning Algorithms

3.1 Memoryless Learning

This algorithm assumes that the learner starts with a randomly chosen grammar (from the n available grammars). He then receives b sample sentences in succession from the teacher (in our case, the parent). For each sentence, if it is consistent with the learner's current grammar, there is no change; otherwise,the learner randomly picks a different grammar from the available set. Given that the overlap between any two distinct grammars is given by a, we can compute the learning accuracy, q, for a memoryless learner in terms of n, b and a (see Eq. (27) of [2]).

3.2 Batch Learning

This algorithm is at the opposite extreme from memoryless learning: it assumes effectively infinite memory capacity. The learner listens to b sentences uttered by the teacher, and then has to choose a grammar which is consistent with all of them. If there is more than one candidate grammar, then one is picked uniformly at random. It turns out that for this algorithm, knowledge of pairwise overlap between grammars is insufficient to compute q: we also need to know the intersections between all combinations of three or more grammars. However, if we assume a random configuration of grammars, an expression for q can be derived; this is given by Eq. (34) of [2]. Batch learning turns out to be much more efficient than memoryless learning, as one might expect: for a population of batch learners, the critical value of b such that q is high enough to lead to a one-grammar solution (i.e., $q \geq q_2$) grows with the number of grammars n as $\log n$; whereas it grows in proportion to n for memoryless learners.

3.3 Finite Memory Learning

Clearly, the actual human learning process lies somewhere between the two extremes mentioned above. Thus, we propose using a linear combination of the values of q obtained in the two cases in order to give a more realistic estimate of learning efficiency:

$$q = (1-\theta)\left[1 - \left(1 - \frac{1-a}{n-1}\right)^b \frac{n-1}{n}\right] + \theta\left[\frac{1-(1-a^b)^n}{a^b n}\right] \qquad (3)$$

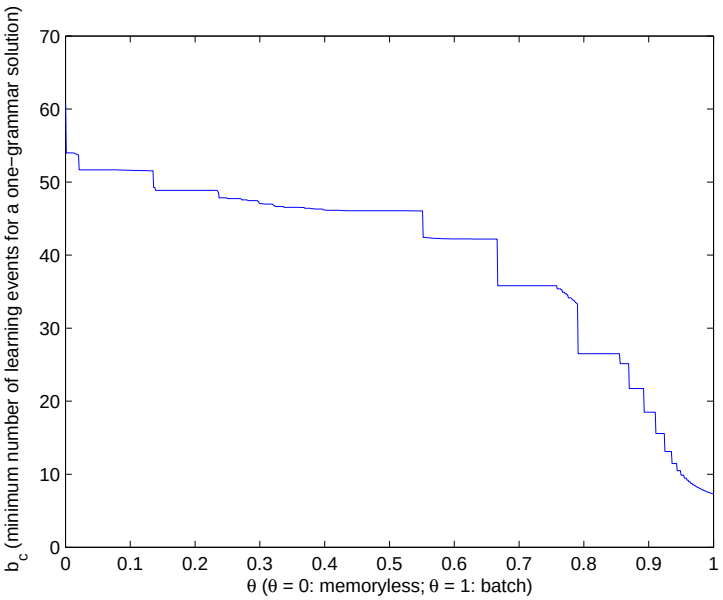

Figure 1: Variation of b_c with change in θ. Parameters used: $a = 0.5$, $f_0 = 1$ and $n = 10$.

Here θ is a parameter which determines the 'quality' of the learner's memory: $\theta = 0$ corresponds to memoryless learning, whilst $\theta = 1$ corresponds to batch learning. In order to study numerically how the critical value of b for the evolution of grammatical coherence (which we denote here by b_c) changes as we vary θ, Equation (3) was coded into Matlab, and the roots of $q(b) = q_2$ were computed for a range of values of θ, using the function $fsolve()$. The other parameter settings used were $a = 0.5$, $f_0 = 1$ and $n = 10$. The results are shown in Figure 1: as expected, the number of sentences needed falls with increasing θ, though a fairly high value of θ is needed before efficiency increases substantially from memoryless learning.

4 Introducing Spatial Variation

The model of [2] does not account for any spatial variation. We extended the model by assuming that the grammatical profile could vary along one spatial dimension. With a spatially varying distribution, we need to consider how to appropriately re-define our fitness function. It is reasonable to assume that each individual will interact only with the population within a small neighbourhood of his own location: so we would like to take some form of weighted average of the grammatical distribution over this neighbourhood, and use that to define the individual's fitness. This suggests the use of the second derivative in space: a positive second derivative at a point implies a higher average value in the neighbourhood, whilst a negative one implies a lower average value. So, denoting our single spatial variable by s, we propose the following modified fitness

function for the fully symmetric case (compare Eq. (3) of [2]):

$$f_i(s,t) = f_0 + a + (1-a)\left(x_i + D\frac{\partial^2 x_i}{\partial s^2}\right) \tag{4}$$

Here f_i denotes the fitness of an individual who speaks the i^{th} language: it is now a function of location in both space and time, as is x_i, the fraction of the population speaking the i^{th} language. Clearly, we can not add the second partial derivative in space of x_i to it directly, because the value of the term may then go out of the $[0,1]$ range, making it meaningless. Thus, it needs to be appropriately scaled, and we multiply it by an unknown constant D: we will see later that this constant will have to be set according to the granularity of the spatial mesh used for numerical integration.

Plugging the fitness function from Equation (4) into the original system, we get (compare Eq. (2) of [2]):

$$\frac{\partial x_j(s,t)}{\partial t} = \sum_{i=1}^{n}\left[f_0 + a + (1-a)\left(x_i + D\frac{\partial^2 x_i}{\partial s^2}\right)\right] x_i Q_{ij} - \phi x_j,\ 1 \le j \le n \tag{5}$$

For the symmetric system, $Q_{ii} = q$ (the learning accuracy) and $Q_{ij} = \frac{1-q}{n-1}$ for $i \neq j$. Also, the last term is added on in order to ensure conservation of the population size, as before: $\phi = \sum_{i=1}^{n} f_i x_i$ is in fact the average fitness of the population, and is termed the *grammatical coherence*.

In order to numerically solve the System (5), we used the method of lines, which explicitly discretizes the spatial derivative, using central differences. Thus, we create a spatial mesh of size S, with a gap size of Δs. We use x_j^s to denote the prevalence of grammar j at point s in the mesh, for $s = 1, 2, ..., S$. Thus, the system becomes:

$$\frac{dx_j^s(t)}{dt} = \sum_{i=1}^{n}\left[f_0 + a + (1-a)\left(x_i^s + D\frac{x_i^{s+1} - 2x_i^s + x_i^{s-1}}{(\Delta s)^2}\right)\right] x_i^s Q_{ij} - \phi^s x_j^s \tag{6}$$

Note that we can now determine what the value of D should be: in order to obtain the meaning of taking a weighted average of the neighbourhood as we had originally intended, we should set $D = \frac{(\Delta s)^2}{4}$. This causes the term inside the fitness function to become $\frac{x_i^{s+1} + 2x_i^s + x_i^{s-1}}{4}$, which is appropriate.

The System (6) was coded in Matlab and integrated using the solver *ode*113(), for a variety of initial conditions. Figures 2 and 3 depict the results for two representative cases. The initial conditions are shown in the upper graphs, and the final steady states in the lower graphs (the accompanying movies show the evolution of both systems in time). In Figure 2, the system has two grammars, with initial conditions given by $x_1(s) = s$ and $x_2(s) = 1 - s$. In Figure 3, there are three grammars, initially distributed as $x_1(s) = 1 - e^{\sqrt{s}-1}$, $x_2(s) = 2s(1-s)$ and $x_3(s) = e^{\sqrt{s}-1} - 2s(1-s)$. The results shown were obtained with the memory parameter θ set to 1 (i.e., batch learning), but the same results were obtained with lower values of θ going down to 0, with the required number of

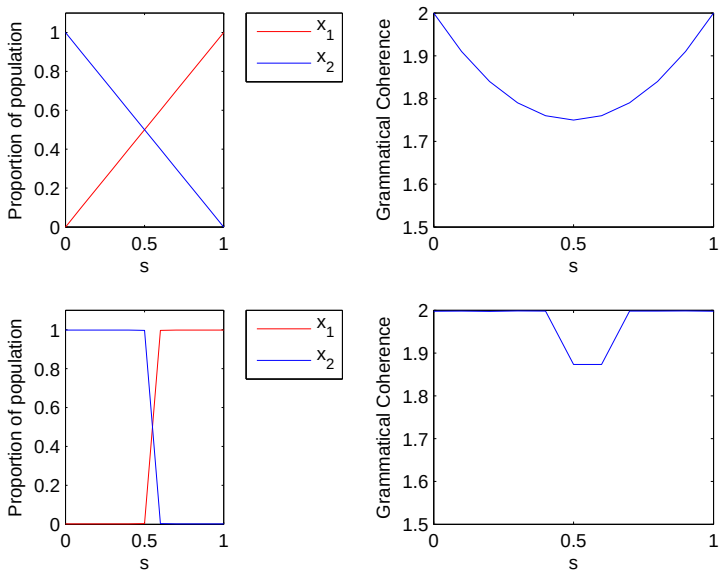

Figure 2: Evolution of a two grammar system with spatial variation. Upper graphs show the initial conditions, and lower graphs are the final steady state. Parameter settings used are $f_0 = 1$, $a = 0.5$, $b = 10$, $\Delta s = 0.1$ and $\theta = 1$ (batch learning).

learning events b increased accordingly. For values of b below a certain threshold, the system converged to a uniform distribution of grammars throughout the spatial domain, in line with our earlier analysis.

The key thing to note about the results is that the systems converge to regions with one-grammar solutions, with fairly well-defined boundaries between these regions. Essentially, the grammar with the maximum initial prevalence in each region becomes dominant there, and all other grammars die out. This maximizes the grammatical coherence within each region; naturally, the coherence dips somewhat at the boundaries. These results correspond well with what we see in the real world: largely homogeneous linguistic regions with sharp boundaries separating them. For instance, all the Romance languages (French, Spanish, Italian, Portuguese and Romanian) probably started out as slightly differing versions of Latin with fairly continuous distributions. Over time, the regions in which each of these was spoken became increasingly well-defined, with minimal overlap between them; and eventually the boundaries became international borders.

5 Conclusions

We have shown that the model of Komarova *et al.* [2] can be quite simply extended to include spatial variation in grammatical profiles, with meaningful results. We have also proposed a new function for learning accuracy which is a

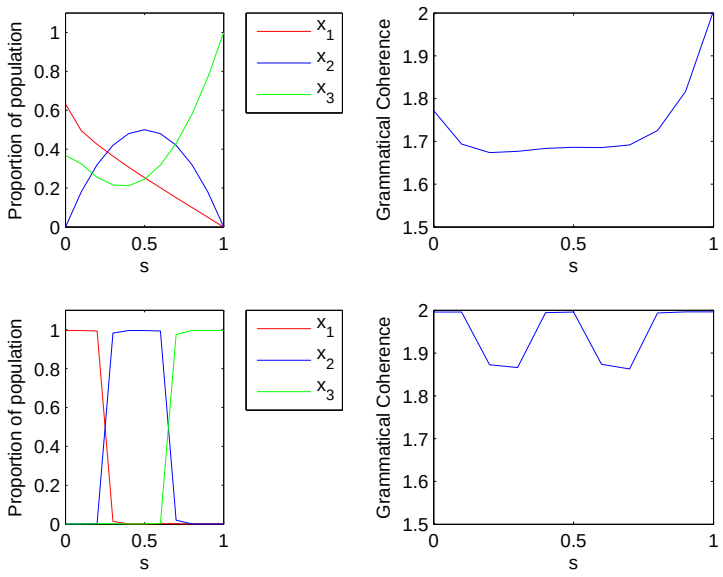

Figure 3: Evolution of a three grammar system with spatial variation. Upper graphs show the initial conditions, and lower graphs are the final steady state. Parameter settings used are $f_0 = 1$, $a = 0.5$, $b = 10$, $\Delta s = 0.1$ and $\theta = 1$ (batch learning).

linear combination of the memoryless and batch learning approaches, and have shown that the minimum number of learning events needed for the emergence of a grammatically coherent population decreases continuously with increase in memory capacity. Both these additions seemingly contribute to making the model more realistic. However, there is no doubt that it is still a drastic oversimplification of language evolution in the real world; and we probably need much greater understanding of the cognitive foundations of language, along with much improved empirical techniques for measuring how grammars really evolve, before we can seriously begin to build more powerful mathematical models.

References

[1] Noam Chomsky. *The Minimalist Program*. The MIT Press, Cambridge, MA, USA (1995).

[2] Natalia L. Komarova, Partha Niyogi and Martin A. Nowak. The Evolutionary Dynamics of Grammar Acquisition. *Journal of Theoretical Biology* 209(1): 43-59 (2001).