Exploring the dynamics of language change in finite populations

By
James BonTempo

Submitted to
Divisional Master's Program in the Physical Sciences
University of Chicago

Advisor
Professor Partha Niyogi

TABLE OF CONTENTS

INTRODUCTION

Taking cues from population biology and the intuition expressed by Lightfoot (1991), Niyogi and Berwick (1995, 1997) introduced a formalized dynamical systems approach to the study of language change based on theories of language acquisition, and demonstrated how the behavior of an individual learner effects the emergence of characteristics on the global level. In particular, it was shown that any combination of grammatical theory, learning algorithm, and sentence distribution gives rise to a dynamical system which allows one to conduct linear stability analysis, generate phase plots, and discover chaotic regimes. This process is demonstrated using variants of the Triggering Learning Algorithm (TLA) proposed by Gibson and Wexler (1994). Niyogi (2002) extends the earlier work by adding analyses of two additional algorithms: the Batch-Error and Cue-Based (Lightfoot, 1999) learning algorithms, and shows how different choices of maturation time and sentence ambiguity, or maturation time and cue probability and threshold, generate a variety of dynamical maps.

However, this framework includes the following problematic assumptions: population sizes are infinite and learners receive input samples, or training data, from the entire population. These have served as the motivations for the present work that:

1. Presents a simulation software package, ALingua[1], which has been developed to model a more realistic situation: finite population sizes with networks[2] defining the source of primary linguistic data (PLD).

2. Explores the effects of the constraints of finite population size and local connection networks on the evolution of a two-language, discrete-time dynamical system.

3. Considers the relationship between spatial distributions and the emergent behavior of the dynamical system.

[1] ALingua can be downloaded at http://alingua.finitestate.net.

[2] Each network is essentially a social network defining the agents with which a single language learner potentially has interactions, and in particular interactions in which the learner is presented with linguistic input.

The results obtained by examining the behavior of the finite system within the stability regimes discovered in the infinite model suggest that the results from the infinite model serve as good approximations to the finite case when considering completely connected populations. However, when the source of PLD is constrained to within a small radius of the learner the qualitative dynamics, and in some cases the quantitative outcomes, of the system are affected. In addition, there are some situations in which a non-random initial distribution of agents also affects the behavior of the system. In the TLA-based simulations there is a decrease in the overall variability when the connection network is localized and an initial period of heightened stability when the initial distribution is non-random, but the long-term behavior is essentially unchanged. The Batch-Error simulations also reflect a variability reduction under a localized PLD source with the long-term behavior being dependent on the distribution of language speakers. Finally, the simulations based on the Cue-Based algorithm suggest reduced variability and also a dependence on the distribution of agents. However, the behavior of the Cue-Based model is significantly different than that observed with the TLA and Batch-Error algorithms. Contrary to the TLA and Batch-Error models, the reduction in variability leads to a reduction, rather than an increase, in the long-term viability of the language under consideration and it is the distribution of the non-cue-driven language that affects the qualitative dynamics and quantitative outcome of the system.

In essence, the ALingua software and its accompanying model embody the last half-century of work in Computational Linguistics; from Chomsky’s pioneering work on Formal Languages in the 1950’s and Gold’s development of a Language Learnability criteria in the 1960’s, to the more recent work of the last decade in both Statistical Learning Theory and the application of Dynamical Systems Analysis to the study of language evolution. What follows immediately is a brief introduction to each of these topics. Next, a summary of the infinite model and the results from Niyogi (2002) are given. Then the particulars of the finite model and the results obtained by examining its behavior are specified. Finally, the conclusions drawn from the present work are offered along with suggestions for further work.

FORMAL LANGUAGE THEORY[1]

The first step in defining language is the establishment of a finite alphabet of distinct symbols Σ. This alphabet can be any such set, for example the set $\{0,1,...,9\}$ when considering decimal numbers, or the set of ASCII characters when considering digitized text. Many times it is useful to consider the binary alphabet {0, 1}[2]. A string s of length k, s^k, is then simply a sequence constructed by concatenating k symbols from the chosen alphabet, and the set of all possible strings over the alphabet is denoted by $\Sigma^* = \bigcup_{k \geq 0} \Sigma^k$. Subsequently, a language L is defined as a subset of all possible strings; that is $L \subseteq \Sigma^*$.

In the natural language setting, one often considers the alphabet consisting of the set of all phonemes, or perhaps the set of words in a language, or vocabulary V, also known as the lexicon. A language can then be defined, for example, as a set of sentences constructed using words from the vocabulary, $L \subseteq V^* = \bigcup_{k > 0} V^k$.

It is then possible to construct a grammar G for the (possibly infinite) language L, such that $\mathrm{L}_G = L$, which is a finite[3] collection of *rewrite rules* that define the set of well-formed expressions in the language. These rules are strings that contain terminals, or symbols from the alphabet Σ, and non-terminals that are variables that can be *expanded* to strings of terminals. Figure 1 shows an example of a very simple

[1] The following conventions have been used in this section and those that follow: L is a language, G is a grammar, M is a Turing Machine, L_G is the language generated by grammar G, and $\mathrm{L}(M)$ is the language accepted by Turing Machine M.

[2] In fact, any alphabet can be represented in the binary alphabet by a process of enumeration.

[3] It is the finite nature of grammars that is of most interest in the current context, as it seems reasonable to imagine that a language learner would hypothesize this structure rather than the possibly infinite set of sentences of the target language.

grammar that generates well-formed sentences[1]. In this grammar, S is a special *start* or *sentence* symbol, the $NP, VP, PP, Noun, Verb$ and $Prep$ symbols are non-terminals, and *time*, *flies*, *arrows*, and *like* are terminal symbols. It should be noted that while a grammar uniquely defines a language, there are potentially infinitely many grammars that can generate the same language. The trivial case would include grammars that have additional *dummy* rewrite rules that are a part of the grammar but are never accessed.

Finally, there are various classes of abstract machines M that can be constructed that accept/parse the strings of a given language L, where $\mathrm{L}(M) = L$. The simplest of these, an example of which is shown in Figure 2, is the Finite State Automata (FSA). Formally, it is a 5-tuple $(\Sigma, Q, q_0, F, \delta)$ where Σ is the input alphabet, Q is a finite set of states, $q_0 \in Q$ is the start state, $F \subseteq Q$ is a set of final states, and δ is a transition function mapping states and input symbols to states, $\delta : Q \text{ x } \Sigma \to Q$. Next is the Pushdown Automata (PDA) that is similar to the FSA but has access to a *stack* where it can store a potentially unbounded amount of information. A PDA is a 7-tuple $(\Sigma, \Gamma, \wedge, Q, q_0, F, \delta)$ where Γ is the stack alphabet, $\wedge \in \Gamma$ is the initial stack symbol, δ is a transition function mapping states, input symbols, and stack symbols to states and stack symbols, $\delta : Q \text{ x } \Sigma \text{ x } \Gamma \to Q \text{ x } \Gamma^*$, and the remaining parameters have the same definitions as they do for the FSA. Last, the Turing Machine has access to a read/write tape[2] and is a 9-tuple $(\Sigma, \Gamma, \text{▯}, \text{ò}, Q, q_0, t, r, \delta)$ where Γ is the tape alphabet containing Σ as a subset, $\text{▯} \in \Gamma - \Sigma$ is the left end marker, $\text{ò} \in \Gamma - \Sigma$ is the blank symbol, $t \in Q$ is the accept state, $r \in Q$ is the reject state, δ is a

[1] Notice that this is a purely syntactic approach to language. That is, it is possible to generate well-formed sentences from this grammar that are not semantically correct or meaningful. For example, while it can produce the phrase, “time flies like arrows,” which describes the passing of time, since both “time” and “flies” can be either nouns or verbs, and “like” can be either a verb or a preposition, it is also possible to interpret the phrase as referring to a particular type of flies that enjoy arrows. Obviously, there are other possible meaningless semantic interpretations, as well. The canonical example of a well-formed sequence that is semantically ambiguous, attributable to Chomsky, is, “colorless, green ideas sleep furiously.”

[2] The standard Turing Machine has an infinite tape, but there are various well-known modifications to this construction. One particularly useful version is the Linear-Bounded Turing Machine which has a finite tape of length kn, where n is the size of the input and k is a constant associated with the machine.

transition function mapping states and tape symbols to states, tape symbols and direction (i.e. left or right), $\delta : Q \text{ x } \Gamma \rightarrow Q \text{ x } \Gamma \text{ x } \{L, R\}$, and the remaining parameters are defined as above.

It can be shown that there is an equivalence relationship between certain classes of languages, grammars, and machines, such that $L = \mathrm{L}_G = \mathrm{L}(M)$ [1]. Shown in Figure 3 is the hierarchy[2] introduced by Chomsky (1956) that organizes these by increasing order of complexity. In addition, each successive language contains, as a proper subset, those below it.

Ever since the introduction of the hierarchy, there has been disagreement as to the placement of the set of natural languages within it. The set of grammatically well-formed sentences of English that are constructed using a center-embedding process can be shown to be non-regular, and so provides simple evidence suggesting that $\mathrm{L}_{English}$ is not in the Type 3, or Regular category. First, construct the set S_{embed} containing sentences of the form $NP^i\ VP^i$, e.g. "The boy yawned," "The boy the man scolded yawned," and so on. This set is demonstrably non-regular by application of the Pumping Lemma. That is, if we assume that S_{embed} is regular, then $\exists\, n > 0$ such that for any string $s = xyz \in S_{embed}$, where $|s| > n$, it should also be the case that $xy^i z \in S_{embed}\ \forall i > 0$, with $|y| > 0$ and $|xy| < n$. However, it is easy to see how any choice for sub-string y leads to an ill-formed string, so S_{embed} is not regular. Next, we observe that $S_{embed} = \mathrm{L}_{English} \cap NP^*\ VP^*$. The set $NP^*\ VP^*$ is regular, so if we assume that $\mathrm{L}_{English}$ is regular S_{embed} must also be regular (since regular languages are closed under intersection). However, we have shown that S_{embed} is not regular, so by contradiction we see that our assumption that $\mathrm{L}_{English}$ is regular was false. In fact, it is not uncommon to place the set of natural languages in the vicinity of the Type 2, or Context-Free category. An example of an argument to the contrary is presented in Kornai (1985) which

[1] See Hopcroft and Ullman (1979) or Kozen (1997) for proofs of these equivalences.

[2] This is now known simply as the Chomsky Hierarchy.

places natural language string sets in the Type 3, or Regular category. As will be shown later, the placement of natural languages within this hierarchy has implications to their learnability.

LANGUAGE LEARNABILITY

Amazingly, children learn their native language through a process of inductive inference; each child is presented with primary linguistic data from a target language and from these a language is hypothesized. This feat is particularly impressive when considering that the learner is exposed to primarily positive examples and is never explicitly given the rules of production[1]. Furthermore, Brown and Hanlon (1970) discovered that learning proceeds with rare correction of ungrammatical utterances[2] and equally likely fulfillment of both grammatical and ungrammatical requests.

Abstractly, we can consider a language learner to simply be an effective procedure, or algorithm A , that maps example sentences to a set of possible hypothesized grammars H , which for the present purposes can be assumed to be equal to L , the set of natural languages. If a sequence of k sentences is denoted by $(s_1, s_2, ..., s_k)$, where $s_i \in \Sigma^*$, we can define the set of all possible sequences of k sentences $D^k = (\Sigma^*)^k$. Furthermore, if the set of all such finite sequences is $\mathrm{D} = \bigcup_{k>0} D^k$, A is then the partial recursive function $\mathrm{A} : \mathrm{D} \to \mathrm{H}$.

Gold (1967) developed a framework for investigating language learnability - language identification in the limit. A language is defined as a set of strings over a finite alphabet and a text for a language is a sequence of strings (in any order) from the language such that each possible string of the language occurs

[1] It is interesting to note that these rules elude the teacher, as well. That is, there is never an explicit understanding of the underlying grammar.

[2] The exception to this generalization is untrue statements.

at least once. Within this context, a language is learnable if there exists a learning algorithm which, when presented with all texts of the language[1], always correctly identifies/converges to the target.

Definition 1 *A target language* L *is learnable if* $\exists$ *an algorithm* A *s.t.* $\forall$ *texts* t *for* $L, \exists n$ *s.t.* $\forall k > n$, $\mathrm{L}_{\mathrm{A}(t_k)} = L$.

Additionally, Gold defines what it means for a class of languages to be learnable.

Definition 2 *A class of languages* L *is learnable if* $\forall L \in \mathrm{L}$ L *is learnable.*

Gold discovered that the set of all finite languages is learnable from a text, but that super-finite sets[2] are not. In fact, under this framework, the entire Chomsky Hierarchy is unlearnable[3]. Some have incorrectly interpreted this result to imply that the set of natural languages is unlearnable. However, the correct interpretation is that the learnability of natural languages is dependent upon the set being constrained. More poignantly, the conclusion can be understood as evidence of the existence of the Universal Grammar (UG) proposed by Chomsky, or that the learner has some form of innate *a priori* knowledge that assists them in the language acquisition process.

Application of statistical learning theory (Vapnik, 1998) relaxes the perfect learning constraints of Gold and allows for a high probability of correct identification within an acceptable level of generalization error[4], ϵ, providing a more plausible context for language acquisition. As a further distinction, sentences are drawn in i.i.d. fashion from a distribution that has support on the set of all languages; so, both positive

[1] Since texts for a language are *fair*, in the sense that they are constructed using only strings allowable in the language being considered, they contain only positive examples.

[2] A set containing all finite languages and at least one infinite language is known as a super-finite set.

[3] One might be tempted to think that the class of regular languages is learnable, however since this category contains all finite languages and also many infinite languages, it too is unlearnable.

[4] This is referred to as *Probably Approximately Correct* learning.

and negative examples are presented. In this framework, a language is learnable if, in the limit, the probability that the distance[1] between the hypothesized and target languages is within the desired confidence interval goes to zero.

> **Definition 3** *A target language L is learnable if $\exists$ an algorithm* A *s.t.* $\forall \epsilon > 0, \Pr[d(\mathrm{L}_{\mathrm{A}(n)}, L) > \epsilon] \to 0$ *as $n \to \infty$, where n is the number of sample sentences and d is the distance between the hypothesized and target languages.*

A family of languages is then learnable in the same sense as above. Application of the Hoeffding bound yields an estimation of a finite number of sentences after whose presentation there is a high probability that the algorithm's guess is within an ϵ-approximation of the target. Interestingly, while this approach effectively extends the class of learnable languages, it ultimately affirms the same Gold-inspired conclusion that learnability necessitates a constrained set of hypothesis languages.

LANGUAGE CHANGE

Language learning is decidedly imperfect. If it were not, languages would be transmitted from generation to generation without modification. This is clearly not the case. Is it reasonable to conjecture that faulty language acquisition, or misconvergence, could give rise to fluctuations in the linguistic composition of a population over time?

Consider what might give rise to change at the individual level. If both a grammatical theory and learning algorithm have been fixed, there are essentially two possibilities, both of which can be seen as

[1] If $1_{L_i}(s)$ is an indicator function defining membership of sentence $s \in \Sigma^*$ in language L_i and P is a probability distribution on Σ^*, then $d(L_1, L_2) = \Sigma \left| 1_{L_1}(s) - 1_{L_2}(s) \right| P(s)$

obvious consequences of the statistical learning context described above, where perfect convergence is unnecessary and in which the number of sample sentences is intimately linked to the probability that the learner is within the required ϵ-approximation of the target language. First, since the learner is being exposed to examples from the set of all languages[1], it is entirely possible that the learner will subsequently fail to adequately approximate the target language, and may even converge to another, as the result of being exposed to too many positive examples from different languages. It also seems natural to draw an association between the Hoeffding bound on the number of sentences and the learner's maturation time, or the time after which a grammatical hypothesis becomes solidified, and thus expose a second possible explanation for misconvergence. In particular, if a learner has reached maturation, yet has not been presented with a sufficient number of unambiguous[2] sample sentences, the learner will have not acquired a language within the tolerable error level of the target.

Against this background of individual misconvergence, it is easy to see how changes would be reflected on the population level. Take the simplest possible case: a homogenous population, i.e. one in which everybody speaks the same language. Due to the inherent stochastic nature of the learning process, it is possible for members of a new generation of learners to hypothesize a language other than the target, and so there exists a non-zero probability that the linguistic composition of the population will change. If the members of a successive generation are then presented with input primarily from those who have earlier misconverged, they will either acquire this new language, or even a completely different one, once again due to the stochasticity in the system. It then becomes obvious how behavior at the individual level gives rise to global change.

However, if imperfect learning is in fact the source of variation, the disparity between the set of possible target languages and that actually acquired by the learner can not be too great otherwise there

[1] In this context, the possible sources of the negative examples are many, including exposure to foreign speakers (an individual or an entire population), non-native speakers, or even those with speech disorders.

[2] An unambiguous sentence is one that can only be parsed/generated by a single grammar in the hypothesis set.

would be a marked decrease in the linguistic coherence of the system[1]. On the contrary, there must be an allowance for a sufficient amount of error to account for the historically observed changes. So, we see that there is an intrinsic friction between language acquisition and language change, which predictably leads to questions regarding the exact nature of their relationship.

THE INFINITE MODEL

Niyogi and Berwick (1995, 1997) show explicitly how the triple $(\mathrm{G}, \mathrm{A}, P)$ of grammatical theory (e.g. parameterization), learning algorithm, and initial sentence distribution naturally gives rise to a dynamical system. Analysis of the system provides insight into explanations for how and why languages evolve along certain trajectories across generations. In particular, the framework fosters an understanding of how individual behavior leads to emergent, global characteristics. This work is then continued in Niyogi (2002) where the scope of considered learning algorithms is widened.

The model presented therein is infinite in size, with every member of the population being *connected*[2] to each other, and embodies the simplest possible situation: a system of two languages, L_1 and L_2 [3], in competition. It is predicated on a syntactic[4] worldview, where languages are taken to be sets of well-formed sequences over a lexicon constructed from a shared underlying alphabet. In addition, the languages are not necessarily disjoint; it is possible for sentences to be ambiguous and belong to both L_1 and L_2. That is, there may be sentences that can be parsed by both grammars. The population is also monolingual; each agent hypothesizes, and after maturation maintains, only a single grammar. Lastly, generations are coincident and discrete, with all new members of the population

[1] Nowak et al. (2001) cast this in terms of the evolution of a Universal Grammar and determine a *coherence threshold* which bounds the conditions under which grammatical communication can evolve.

[2] Since the population is completely connected, any member can present sample sentences.

[3] In a principles and parameters context, one could interpret these as two languages that differ with respect to a single binary-valued parameter, e.g. Pro-Drop or Verb-Second $(V2)$ movement.

[4] This could easily be extended to a phonological context, where the lexicon is generated from a shared set of phonemes.

maturing and being added at the same time, and the location of each individual is static; there is no migratory movement in the system.

There are a handful of values that are then required to completely encapsulate the dynamical system. The proportion of L_1 speakers is given by α, so at any time the state of the system can be described by α_t, which is the proportion of L_1 speakers at time t (obviously, the proportion of L_2 speakers is $1-\alpha_t$). Also, the probability function P_1 gives the distribution of sentences over L_1, while P_2 is the equivalent for sentences of L_2. Related to the probability functions, and of immense import, are the parameters $a = P_1[L_1 \cap L_2]$ and $b = P_2[L_2 \cap L_1]$ which give the probability with which L_1 and L_2 speakers produce ambiguous sentences, respectively[1]. The now familiar learning algorithm A defines a mapping from sample sentences to the set of possible languages, $\{L_1, L_2\}$, and K gives the number of sample sentences presented to the learner.

The first of the three analyzed algorithms is the Triggering Learning Algorithm (TLA) (Gibson and Wexler, 1994), a memory-less, gradient ascent learning process that manipulates a set of parameters[2]. A slightly modified version of this algorithm is used in this model since the class of hypothesis languages contains only two, L_1 and L_2. Initially, the learner chooses one of the possible languages at random. Upon receiving the first training datum, if the current hypothesized language correctly parses/generates the sample, it is retained and the next sample is processed. However, if the learner cannot successfully parse the sentence, they will change their hypothesis to the other language available in the system. Furthermore, the algorithm is not greedy. The sample sentence is never reanalyzed using the newly posited language; the flip is permanent and the next sample is analyzed accordingly.

The second algorithm considered is the Batch Error Learner. In contrast to the memory-less process of the TLA, the Batch Error procedure waits until all example sentences have been presented and then

[1] All of the probability distributions, once set, are maintained for the entire process.

[2] In the Principles and Parameters tradition, this would correspond to a set of binary-valued parameter settings that uniquely defines a language.

chooses the language that most reliably accounts for the observed data[1]. More specifically, the learner considers three subsets of their primary linguistic data: the unambiguous sentences of L_1 $(L_1 \setminus L_2)$, the unambiguous sentences of L_2 $(L_2 \setminus L_1)$, and the ambiguous sentences $(L_1 \cap L_2)$. If n_1, n_2, and n_3 are the number of sentences in these sets, respectively, then the learner chooses L_1 if $n_1 > n_2$, L_2 if $n_2 > n_1$, or either L_1 or L_2 with some fixed probability if $n_1 = n_2$ [2].

The final algorithm included in the analysis is the Cue-Based Learner as described by Lightfoot (1999). Using this method, the learner analyzes the incoming data looking for *cues*; abstract structures derived from the sample sentences that are taken to be equivalent to the parameters accessed in the TLA model. Each language in the target class is uniquely defined by a set of cues that are found embedded[3] in a subset of its unambiguous sentences, e.g. for L_1 the set of cues $C_1 \subseteq L_1 \setminus L_2$. A cue threshold τ for the target language is then chosen, and if the proportion of cues in the set of sample sentences is greater than the threshold, the learner chooses the target.

RESULTS FROM THE INFINITE MODEL

For the TLA learner, the derived difference equation for a fixed number of finite sample sentences is

$$\alpha_{t+1} = \frac{B + \frac{1}{2}(A-B)(1-A-B)^K}{A+B},$$

[1] The actual algorithm employed by a real language learner most likely lies somewhere between the memory-less and batch error procedures. While the computational cost of the TLA is extremely low, it suffers by not having recourse to previous experience. As for the Batch Error learner, it has access to the entire history of training data, but at a high computational cost.

[2] In the present implementation, if $n_1 = n_2$ the learner chooses L_1 with probability 1.

[3] This approach assumes that the cues are located in degree-0 matrix clauses, and thus not deeply embedded in an utterance.

where $A = (1-\alpha_t)(1-b)$ and $B = \alpha_t(1-a)$. Analysis of this equation yields a single stable fixed point in the closed interval $[0,1]$ to which the system converges given any set of initial conditions. If $a = b$ the stable fixed point occurs at $\alpha_t = \frac{1}{2}$. If $a > b$ stability occurs rather close to $\alpha = 0$ and most of the population speaks L_2. Otherwise, the stable point of the system is close to $\alpha = 1$ and the system will contain L_1 speakers almost exclusively. If the number of sample sentences approaches ∞, the update equation becomes simply

$$\alpha_{t+1} = \frac{\alpha_t(1-a)}{\alpha_t(1-a) + (1-\alpha_t)(1-b)}.$$

Under these conditions, if $a = b$ then $\alpha_{t+1} = \alpha_t$ and the initial proportions are maintained. If $a > b$ then there are two fixed points in the system: $\alpha = 1$ which is stable and $\alpha = 0$ which is unstable. Otherwise, there are the same two fixed points, but $\alpha = 1$ is unstable and $\alpha = 0$ is now stable.

For the Batch Error procedure, the following difference equation is obtained

$$\alpha_{t+1} = \sum_{n_1, n_2, n_3 | n_1 \geq n_2} \binom{K}{n_1, n_2, n_3} p_1^{\,n_1} p_2^{\,n_2} p_3^{\,n_3}$$

where $p_1 = \alpha_t(1-a)$, $p_2 = (1-\alpha_t)(1-b)$, and $p_3 = \alpha_t a + (1-\alpha_t)b$. If $b = 1$ then $p_2 = 0$ and n_2 will always be zero, so $\Pr[n_1 \geq n_2] = 1$ and $\alpha = 1$ is a stable fixed point. If $b \neq 1$ and $a = 1$ then $p_1 = 0$, so $\Pr[n_1 \geq n_2] = \Pr[n_2 = 0]$ and the update equation is reduced to

$$\alpha_{t+1} = \left[1 - (1-\alpha_t)(1-b)\right]^K.$$

Consequently, the fixed point $\alpha = 1$ is stable only if $b > 1 - \frac{1}{K}$, otherwise for smaller values of b a different stable point arises in the open interval $(0, 1)$. For most other choices of a and b $\alpha = 1$ is stable, and there are two other fixed points α_1 and α_2 in the open interval $(0,1)$, with $\alpha_1 < \alpha_2$ where α_1 is stable and α_2 is unstable. If the number of sample sentences approaches ∞, $\frac{n_1}{K} \to p_1$ and $\frac{n_2}{K} \to p_2$. Accordingly, $\alpha_t \to 1$ if $p_1 > p_2$, or rather if $\alpha = \alpha_t(1-a) > \alpha_* = (1-\alpha_t)(1-b)$. Clearly, $\alpha = 1$ and $\alpha = 0$ are both stable fixed points and $\alpha = \frac{1-b}{(1-b)+(1-a)}$ is an unstable fixed point between them.

Finally, for the Cue-Based process, the dynamics are given by the difference equation

$$\alpha_{t+1} = \sum_{K\tau \le i \le K} \binom{K}{i} (p\alpha_t)^i (1 - p\alpha_t)^{K-i}$$

where $p = P_1(C)$. If $p = 0$ then the only stable point in the system is $\alpha = 0$, as is also the case for small values of p. A bifurcation occurs as p increases, creating two additional fixed points α_1 and α_2, where $\alpha = 0$ is still stable, α_1 is unstable, and $\alpha_2 > \alpha_1$ is stable. If $p = 1$ then both $\alpha = 1$ and $\alpha = 0$ are stable fixed points with one unstable fixed point between them. If the number of samples sentences approaches ∞, then $\frac{k}{K} \to p\alpha_t$. If $p < \tau$, then $\alpha = 0$ is the only fixed point in the system. Otherwise for all $0 \le \alpha_0 < \frac{\tau}{p}$, $\alpha = 0$ is stable, and for all $\frac{\tau}{p} < \alpha_0 \le 1$, $\alpha = 1$ is stable.

THE FINITE MODEL

In many respects the finite model that has been implemented is the same as the infinite model explored above. It is based on the same syntactic foundation. There are still only two languages in the system[1], L_1 and L_2, that are in competition and may include subsets of sentences that are analyzable by both grammars. Also, the population is monolingual, and the generations are coincident, discrete and non-migratory. Finally, the same three learning algorithms, the TLA, Batch-Error, and Cue-Based learners, have been included[2].

There are three major enhancements in the finite model, whose inclusion is intended to create a more realistic context for examining the dynamics of language change:

> **1**. A finite population that exists within a contained space[3] - It is quite obvious that this more closely resembles the situation of a real language learner in contrast to the infinite population model. In order to avoid problems at spatial boundaries, the surface on which the individuals reside is toroidal, so those at an edge still have a complete set of *neighbors*.
>
> **2**. The capacity to define a social network for each individual that determines the possible sources of primary linguistic data - In the infinite model, every member of the population was *connected* to each other and so the entire population influenced the dynamics of the system. However, in the finite model there is finer control of the population's connectedness. While it is still possible to configure a completely connected population, it is also possible to explicitly delimit a *neighborhood* of individuals that are situated within a certain radius of each leaner. This allows the model to approximate the notion of real-world

[1] Adding the ability to include more languages to the system has been left for future development.

[2] The implementations of the Batch-Error and Cue-Based algorithms in Alingua are slight modifications of those considered in the infinite model analysis. In particular, they allow for the specification of the fixed probability for the Batch-Error process where the infinite model assumed a probability of 1, and the cue probabilities and thresholds for both languages under the Cue-Based approach where the infinite model only considered the effects of the cue probability and threshold for one of the languages in the system.

[3] For display purposes, the maximum population size is 250,000 - a 500 x 500 square. This is more than sufficient for the present investigation.

neighborhoods, which many times constitute the main source of a learner's training data. In addition, it is also possible to construct a completely randomized connection network, or to define a separate, possibly unique network for each learner.

3. The ability to explicitly define the distribution of language speakers when setting the initial conditions of the system - This permits the investigation of the effects of different spatial distributions on the behavior of the system.

It is clear that these three improvements to the original infinite model produce a framework for studying the evolution of a system of competitive languages that more closely approximates the circumstances of language acquisition in the real world. Consequently, the software will serve as a useful tool in the search for viable theories of diachronic language change.

RESULTS FROM THE FINITE MODEL

The stability regimes determined by the analysis of the three algorithms in the infinite model were examined in finite populations using the Alingua software package. A standard population of 10,000 individuals was chosen in order to minimize the background variability due to population size[1]. Also, a maturation time of 128 sentences was used as it can be shown that this number of samples is sufficient to ensure a high probability that a learner will converge to the target language in a 3-parameter system[2]. There were two connection network topologies considered, complete and local (with a radius of 1), and three initial distributions of language speakers[3]: random, island, which is an elliptical cluster of similar language users, and split, which is a rectangular-shaped cluster that extends vertically across the entire

[1] Niyogi (2002) briefly discusses the effects of population size on the variability of the system. In particular, high levels of variance are observed in small populations – an effect familiar to population biologists studying genetic drift and conservation biologists examining the long-term viability of small populations.

[2] Obviously, this is more than sufficient for the current 1-parameter system.

[3] Presumably there would be a total of six combinations of distributions and networks included in the present analysis. However, a completely connected network results in the destruction of any initial distribution after a single generation and so there are only 4 combinations considered: random/complete, random/local, island/local and split/local.

population space. Finally, there are two recurring values in the analysis: the average population change per generation, $\overline{\Delta\alpha} = \frac{1}{n}\sum\left|\alpha_t - \alpha_{t-1}\right|$ where n is the number of realizations, and the average local variance per language which is an overall measure of the clustering of speakers of similar languages, with the average local variance of $L_i = \frac{1}{\left|L_i\right|} \sum_{\substack{A_j | L_{A_j} = L_i \\ A_n \in \{neighbors\ of\ A_j\}}} d(L_{A_j}, L_{A_n})$, where A_j is the agent at the currently selected location j, $\left|L_i\right|$ is the number of individuals in the population speaking language i, and d is a distance measure giving the similarity between languages[1].

Triggering Learning Algorithm (TLA)

The first algorithm considered in the analysis was the TLA. Figures 4-6 show $\alpha_t, \overline{\Delta\alpha},$ and average local variance for several simulations with $a = b = 0.5$. Analysis of the infinite model predicts $\alpha_t = \frac{1}{2}$ as a stable fixed node and we see that this serves as a good approximation for the finite case, under both connection settings (complete and local) and all initial distributions. While the quantitative outcome for each combination of network and distribution settings is similar, their qualitative behaviors are distinct. First, Figure 4 shows that constraining the PLD source to the local network reduces the overall variability in the system. Additionally, Figure 5 shows that local connections also serve to dampen the variability of $\overline{\Delta\alpha}$, or the change between consecutive generations, suggesting a local predictor of the global variability. Also, the graphs of the island and split distributions show evidence of edge effects, where change in the system occurs at the edge of a language-speaking cluster. Initially, the variability in these systems is very low, with the island distribution showing greater variance and more rapid change due to

[1] Since the current system only includes two languages, L_1 and L_2, $d(L_1, L_2) = 1$. In future multi-lingual versions of Alingua, it will be possible to explicitly define similarities between languages.

the fact that a higher proportion of its population resides at the edge of the cluster as opposed to the split distribution. Eventually, however, changes at the edge filter into the cluster at which point the variability is similar to that in the randomly distributed, locally connected system. Finally, Figure 6 shows that a local network fosters clustering: the average local variance in the randomly distributed, completely connected population remains relatively constant, whereas that in the randomly distributed, locally connected system decreases rapidly during the first few generations (screenshots from ALingua revealing this process are shown in Figure 7). Conversely, the highly clustered nature of the island and split distributions decreases, or the average local variance increases, steadily over time, eventually reaching a level similar to that in the randomly distributed case, suggesting a maximum amount of clustering allowed under these parameter settings.

Analysis of the TLA with $a > b$ shows a similar outcome[1]: the infinite model prediction serves as a good approximation of the quantitative outcome of the system, namely $\alpha_t = 0$, with the qualitative behavior differing depending on the nature of the connection network and the initial distribution. The trajectories of α_t in Figure 8 show the positive effect that local connectivity has on the long-term viability of a population of at-risk language speakers. In particular, the time to $\alpha_t = 0$ in the randomly distributed, locally connected system is almost double that in the population of randomly distributed, completely connected individuals. Once again, local connectivity dampens the variability and magnitude of $\overline{\Delta\alpha}$ as shown in Figure 9, and the introduction of the local network has induced cluster formation (Figure 10) in the randomly distributed system, presumably leading to the increased length in time to extinction. Finally, the dynamics of the island and split distributions is strikingly different from that of the randomly distributed populations due to their highly clustered nature and the consequent slow filtration of L_2 speakers into the L_1 community.

[1] The system is essentially symmetric, so the outcomes for the $a < b$ case are similar, with the numbers and effects reversed.

Batch-Error Learner

The next algorithm analyzed using Alingua was the Batch-Error learner. The infinite model analysis predicts $\alpha_t = 1$ as a stable fixed node if $a = 1$ and $b > 1 - \frac{1}{k}$, and this serves as a good approximation for the finite case. As seen with the TLA analysis, the quantitative outcome of the systems is the same, while the qualitative behaviors are distinct, though the differences in this case are less remarkable. Plots of α_t and the average local variance are shown in Figure 11, where it can be seen that the constraint of a locally connected network effectively decreases the time to stability through a more rapid decrease in the local variance. More interestingly, it is possible for a locally connected system to surpass the stable fixed node, $\alpha_t < 1$, even if $b \le 1 - \frac{1}{k}$, and go to $\alpha_t = 1$ due to the clustering effects of the local connections as seen in Figure 12. This behavior is only observed for ambiguous probabilities close to $1 - \frac{1}{k}$, so it is still the case that the predictions of the infinite model are good approximations for this combination of parameters.

Substantially different behavior is observed with the Batch-Error process when considering the stability of the system about α_*, with the infinite model predicting the stability of $\alpha_t = 0$ for $\alpha_0 < \alpha_*$, and $\alpha_t = 1$ for $\alpha_0 > \alpha_*$. Plots for $\alpha_0 = \{0.15, 0.3, 0.45\}$, $a = b = 0.5$ and $\alpha_* = \frac{1}{2}$ are given in Figure 13[1]. For relatively small values of α_0 the infinite prediction holds and the quantitative outcome is the same for both the completely and locally connected populations. The qualitative behavior is different between them, however, with the clustering induced by the local network leading to an increased time to stability. As α_0 approaches α_*, the quantitative outcomes diverge with the completely connected system converging to the stable fixed point $\alpha_t = 0$ while the locally connected population obtains stability at $\alpha_t \,\square\, 0$. Once

[1] The Batch-Error based system is essentially symmetrical and the results for $\alpha_0 > \alpha_*$ are similar to those presented, just simply inverted.

again, this is an effect of the clustering of individuals speaking the same language, and in particular a clustering into a stable shape. In fact, a split-style distribution is stable while an island-shaped distribution leads to an increase in α; both of these behaviors can be easily understood by considering what happens at two neighboring locations on either side of an edge of the cluster. Individuals located at the edge of a split distribution will have a majority of similar-speaking neighbors, as will their dissimilar-speaking neighbor outside the cluster, and since $a = b$ and K is sufficiently large there is a high probability that $n_1 \geq n_2$ and the next individual at that location will acquire the language of their predecessor. As for the island distribution, each individual on the internal side of the edge will either have a majority of similar speakers or equal numbers of similar and dissimilar ones, as will their outside neighbor. Consequently, there is a high probability that the individuals at either location will retain the current language, or that the outside neighbor will acquire the language of the cluster, with the latter outcome being due to the fact that $a = b$, K is sufficiently large, and the fixed probability for ties is 1. Once the language at a cluster-external location flips, the local change can lead to global amplification. So, if the clustering effects of the local network lead to the development of a split or island-style distribution[1] from an initial random distribution, the system will become stable and significantly diverge from the predicted stability point. If we consider the local variance plots for the $\alpha_0 = 0.45$ runs in Figure 14, this is effectively what is happening as the system rapidly produces dense clustering. The ALingua screenshots provided in Figure 15 give an added visual representation of this process.

Cue-Based Learner

The final algorithm considered in the analysis was the Cue-based learner. Once again, the predictions of the infinite model serve as good approximations for the finite system, with the locally connected

[1] In reality, the populations never settle on a perfect split or island distribution, but rather the edges fluctuate between relatively flat (split-style) and rounded (island-style) edges.

populations *lagging* slightly behind the completely connected system. This can be seen in Figures 16 and 17 where the completely connected system has undergone the bifurcation at $p = 0.151$ while the locally connected system does not exhibit the additional fixed point, but does so when p is increased to 0.155. Unlike the previously considered algorithms, the locally connected populations employing the Cue-Based algorithm are sensitive to the distribution of the non-cue-controlled language in the system. In particular, there are no quantitative or qualitative changes in the system as long as the non-cue-driven language remains randomly distributed. However, if a cluster develops, this can lead to significant changes in the behavior of the system as can be seen in Figures 18 and 19 where an initial island composed of 0.005 percent of the population, or 50 L_2 speakers, leads to the disappearance of the additional stable fixed node.

CONCLUSIONS

Niyogi and Berwick (1995, 1997) and Niyogi (2002), working under the assumptions of infinite population size and complete network connectivity, showed that formalized dynamical systems analysis provides a valuable mechanism for the study of diachronic language change. We have seen that in all the cases considered above the predictions of stability based on the analysis of the infinite model serve as good approximations for the quantitative outcomes of completely connected populations in a finite space. However, the inclusion of a constraint on the source of PLD to a local radius of neighbors, or an explicit initial distribution of language speakers, can significantly affect both the qualitative and quantitative behavior of these systems. The most remarkable behaviors are observed in the Batch-Error based simulations where high levels of clustering are induced by the introduction of a local network and the possibility arises for a large divergence from the predicted stability regions. In addition, the instability of the Cue-Based system introduced by the inclusion of an extremely small cluster of non-cue-driven speakers differs greatly from the predictions of the infinite model.

Ultimately, the Alingua software package has proven to be a valuable tool in the analysis of the dynamics of a two-language system with a finite population. Pragmatically, comparisons between the outcome of simulations and empirical results from historical linguistics will facilitate the search for satisfactory theories of diachronic language change. This has been left open as a possible direction for future research. Other possibilities for future work include the addition of multi-lingual support and the ability to explicitly define similarities between languages so that the calculation of the average local variance would continue to provide practical data. Also, as new learning hypothesis are introduced, or the existing ones are modified, their implementations could be added or modified accordingly.

As an unintended benefit, ALingua may also present a useful model for the study of language extinction, a topic that has provoked concern among linguists (Hale, 1992) and has recently received much attention. For example, Sutherland (2003) provides an interesting examination of the status of the world's languages using the system designed for the classification of species extinction risk. The results suggest that a large number of languages are in danger of disappearing and that we are losing languages at rates even greater than those at which we are losing biological diversity. Presumably, simulations in ALingua can provide insight into the possible mechanisms behind language extinction, namely the effects of connectivity and distribution on long-term viability, and the means by which the present decline in diversity may be ameliorated or even countered.

Finally, and more abstractly, if the material and physical understanding of location implicit in the system is replaced with a socially and culturally defined conception of place as advocated by Johnstone (2004), and a Whorfian-style approach to language is adopted, the results obtained may be interpreted in terms of cultural transmission. In particular, the model may suggest how connectivity, local or otherwise, may give rise to global socio-cultural patterns and the outcomes of simulations may provide insight into benefits or detriments of our ever-increasing global village.

REFERENCES

Brown, R. and C. Hanlon. Derivational complexity and order of acquisition in child speech. In J. Hayes (ed) *Cognition and the Development of Language* (Wiley, New York, 1970).

Chomsky, N. Three models for the description of language. I.R.E Transactions on Information Theory **IT-2**(3), 113-124 (1956).

Kornai, A. Natural Languages and the Chomsky Hierarchy. In M. King (ed) *Proceedings of the 2nd European Conference of the Association for Computational Linguistics*, 1-7 (1985).

Kozen, D. *Automata and Computability* (Springer-Verlag, New York, 1997).

Gold, E. M. Language Identification in the Limit. *Information and Control* **10**, 447-474 (1967).

Hale, K. On endangered languages and the safeguarding of diversity. *Language* **68**, 1-3 (1992).

Hopcroft, J. and J. Ullman. *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, Massachusetts, 1979)

Johnstone, B. Place, Globalization, and Linguistic Variation. In C. Fought (ed) *Sociolinguistic Variation: Critical Refelections* (Oxford University Press, Oxford, 2004)

Lightfoot, D. *How to Set Parameters* (MIT Press, Cambridge, Massachusetts, 1991).

Lightfoot, D. *The Development of Language: Acquisition, Change, and Evolution* (Blackwell, Oxford, 1999)

Niyogi, P. and R.C. Berwick. The Logical Problem of Language Change. *AI-Memo 1516*, Artificial Intelligence Laboratory, MIT (1995).

Niyogi, P. and R.C. Berwick. A Dynamical Systems Model for Language Change. *Complex Systems* **11**, 161-204 (1997).

Niyogi, P. *The Computational Nature of Language Learning and Evolution* (The University of Chicago, Unpublished book draft, 2002).

Nowak, M. A., N. L. Komarova and P. Niyogi. Computational and evolutionary aspects of language. *Nature* **417**, 611-617 (2002).

Sutherland, W. J. Parallel extinction risk and global distribution of languages and species. *Nature* **423**, 276-279 (2003).

Vapnik, V. N. *Statistical Learning Theory* (Wiley, New York, 1998).

FIGURES

$$\begin{aligned}
&S \rightarrow NP\,VP \,|\, VP \\
&NP \rightarrow Noun\ NP \,|\, Noun\ PP \,|\, Noun \\
&VP \rightarrow Verb\ NP \,|\, Verb\ PP \,|\, Verb\ NP\ PP \,|\, Verb \\
&PP \rightarrow Prep\ NP \\
&Noun \rightarrow time \,|\, flies \,|\, arrows \\
&Verb \rightarrow time \,|\, flies \,|\, like \\
&Prep \rightarrow like
\end{aligned}$$

Figure 1 A simple grammar that produces syntactically well-formed sequences.

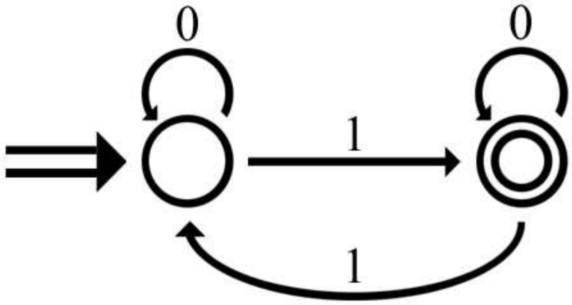

Figure 2 A Finite State Automata that accepts strings from the binary alphabet containing an odd number of ones. Each circle is a state: the double-tailed arrow indicates the start state and the double circle is the final/accepting state. Arcs between the states define the transitions.

	Language	**Grammar**	**Machine**
Type 3	Regular	Regular	Finite State Automata
Type 2	Context-Free	Context-Free	Pushdown Automata
Type 1	Context-Sensitive	Context-Sensitive	Linear-Bounded Turing Machine
Type 0	Recursively Enumerable	Unrestricted	Turing Machine

Figure 3 The Chomsky Hierarchy shows equivalence classes of languages, grammars, and machines, and organizes them by complexity.

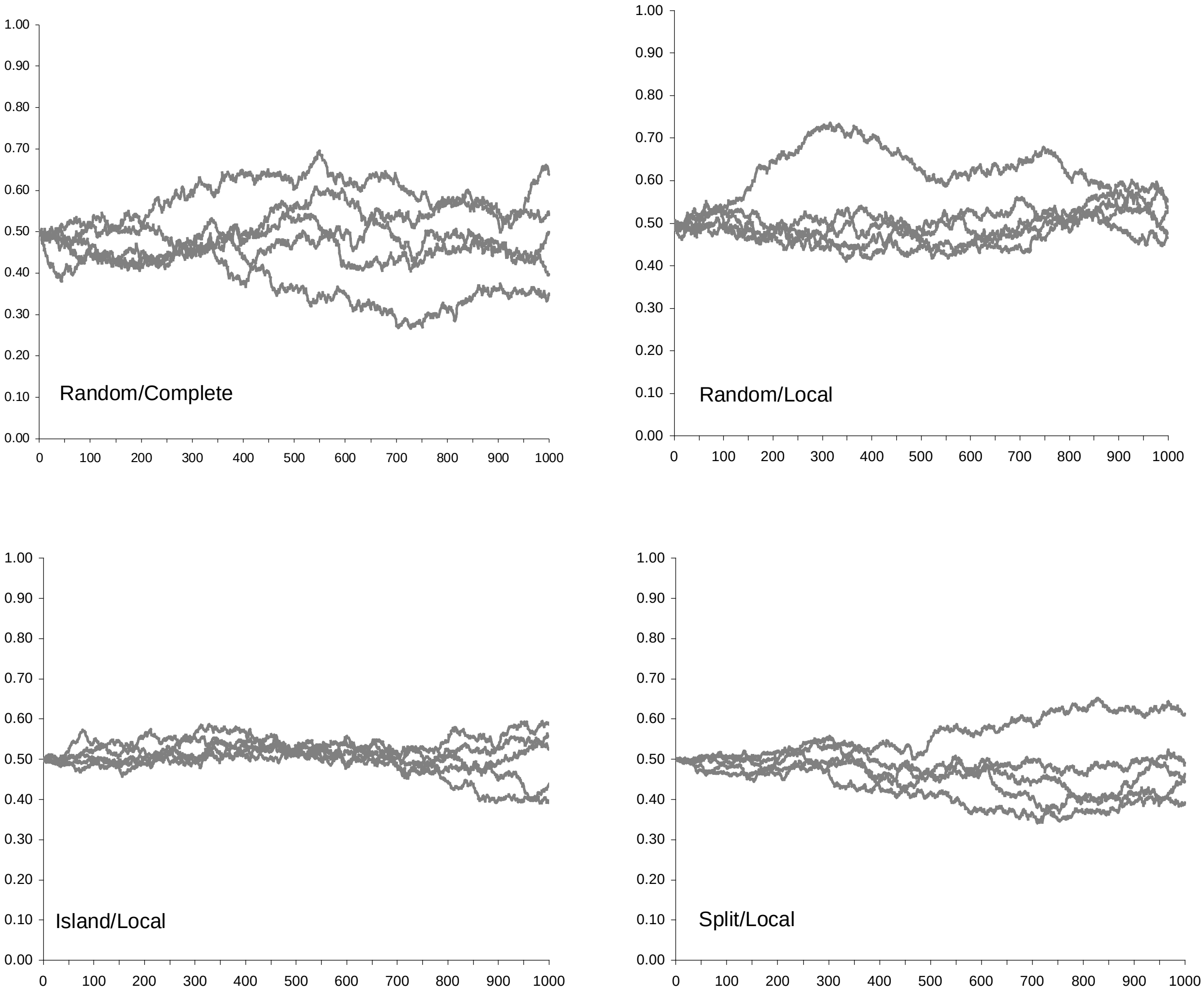

Figure 4: Graphs of α_t over 1000 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = b = 0.5$, and $K = 128$. Notice the significant decrease in variability with the introduction of local network connections. Also, while one run in the random/local configuration is relatively divergent, there is less variation between generations than the completely connected system and the remaining trajectories are tightly clustered.

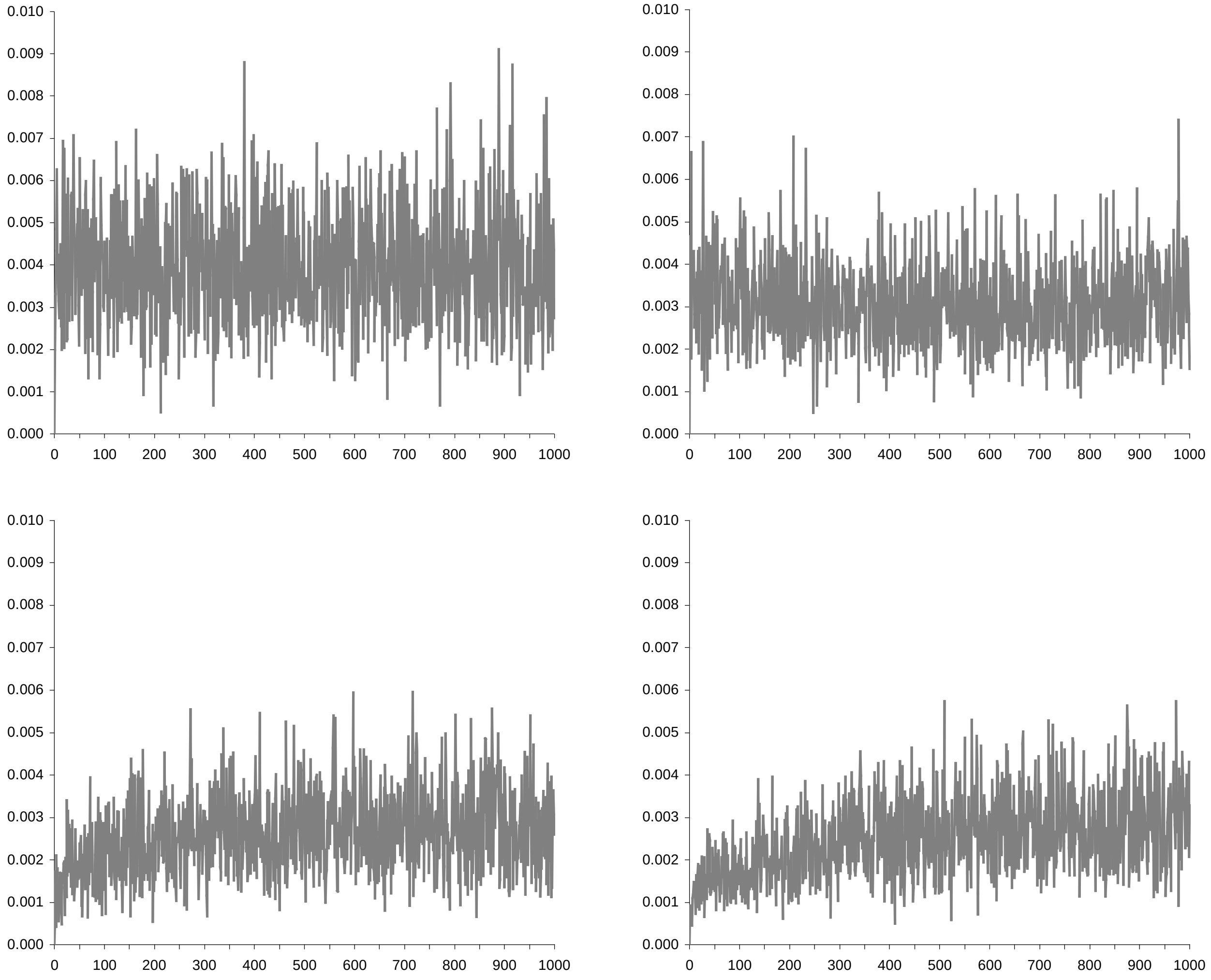

Figure 5: The average generational change of α_t, $\overline{\Delta\alpha}$, over 1000 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = b = 0.5$, and $K = 128$. Notice that the largest variability between generations and greatest average change occurs in the random/complete system. Also, note the low initial values in both the island/local and split/local systems due to their highly clustered nature and edge effects.

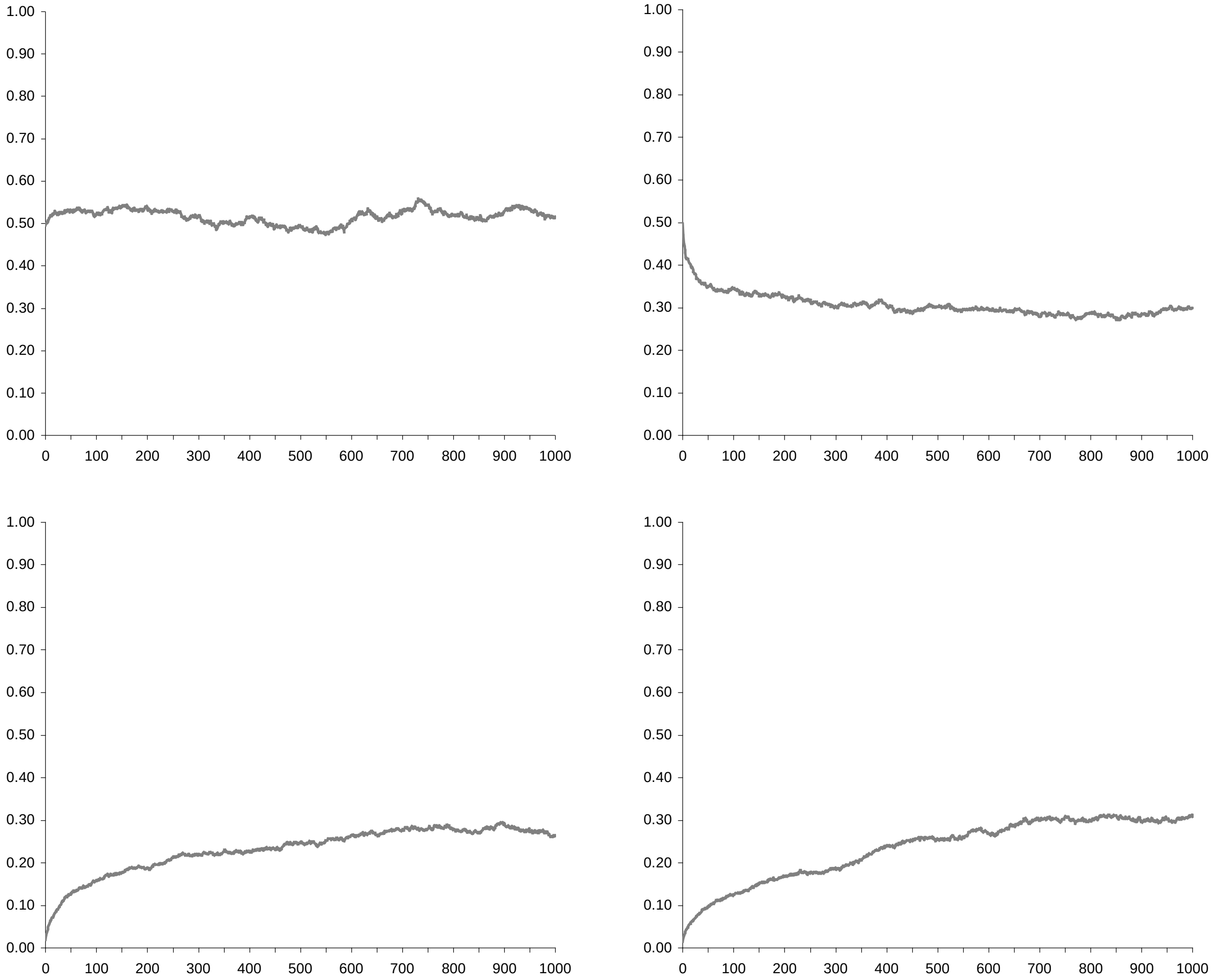

Figure 6: The average local variance over 1000 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = b = 0.5$, and $K = 128$. Notice that the local variance is essentially unchanged in the random/complete system, there is an immediate decrease and leveling in the random/local system, and that both the island/local and split/local systems start at extremely low levels and then increase to a level similar to that of the random/local system.

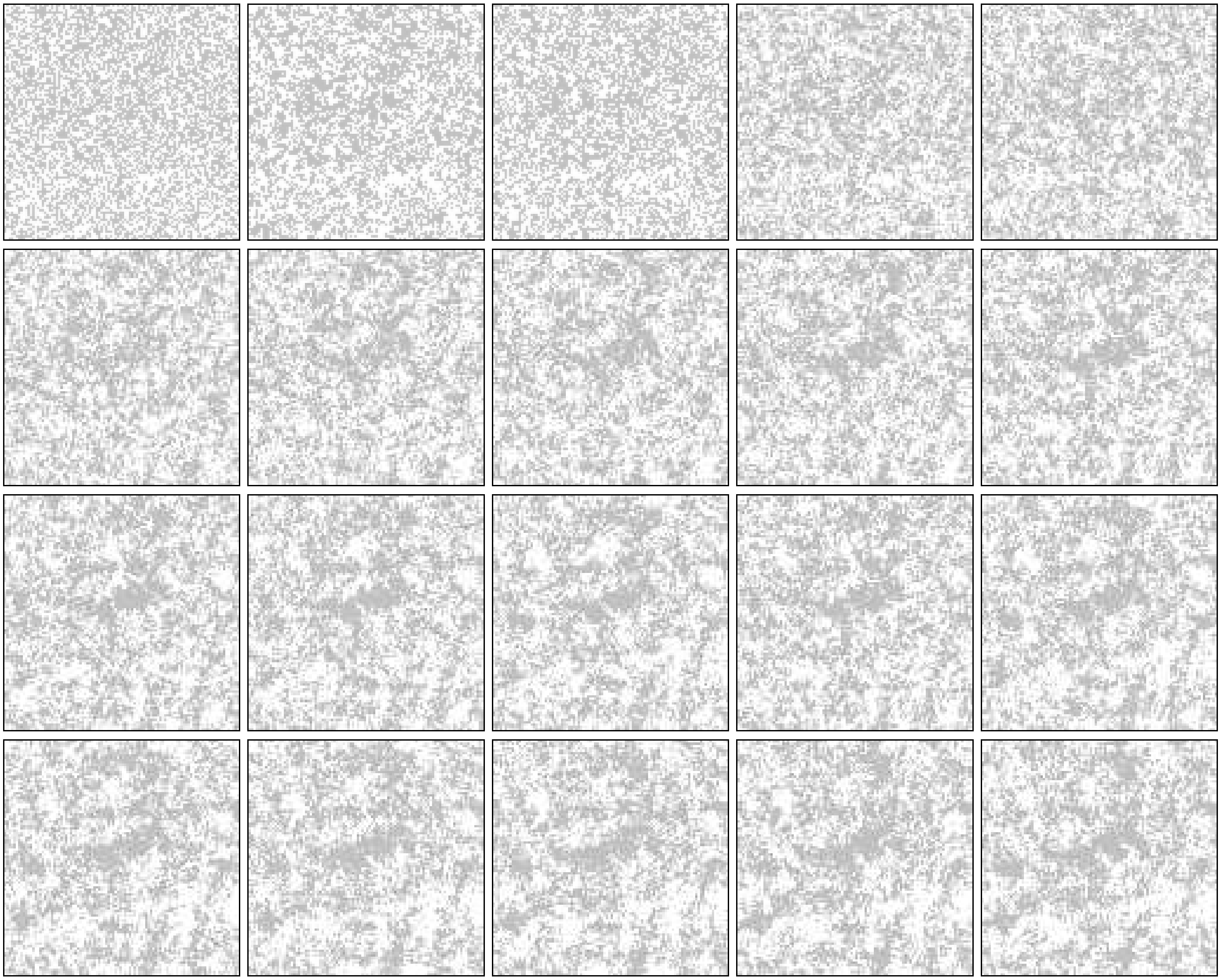

Figure 7: ALingua screenshots of the first 20 generations of a randomly distributed population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = b = 0.5$, and $K = 128$. Notice how the local network connections have induced significant clustering of the initially randomly distributed population

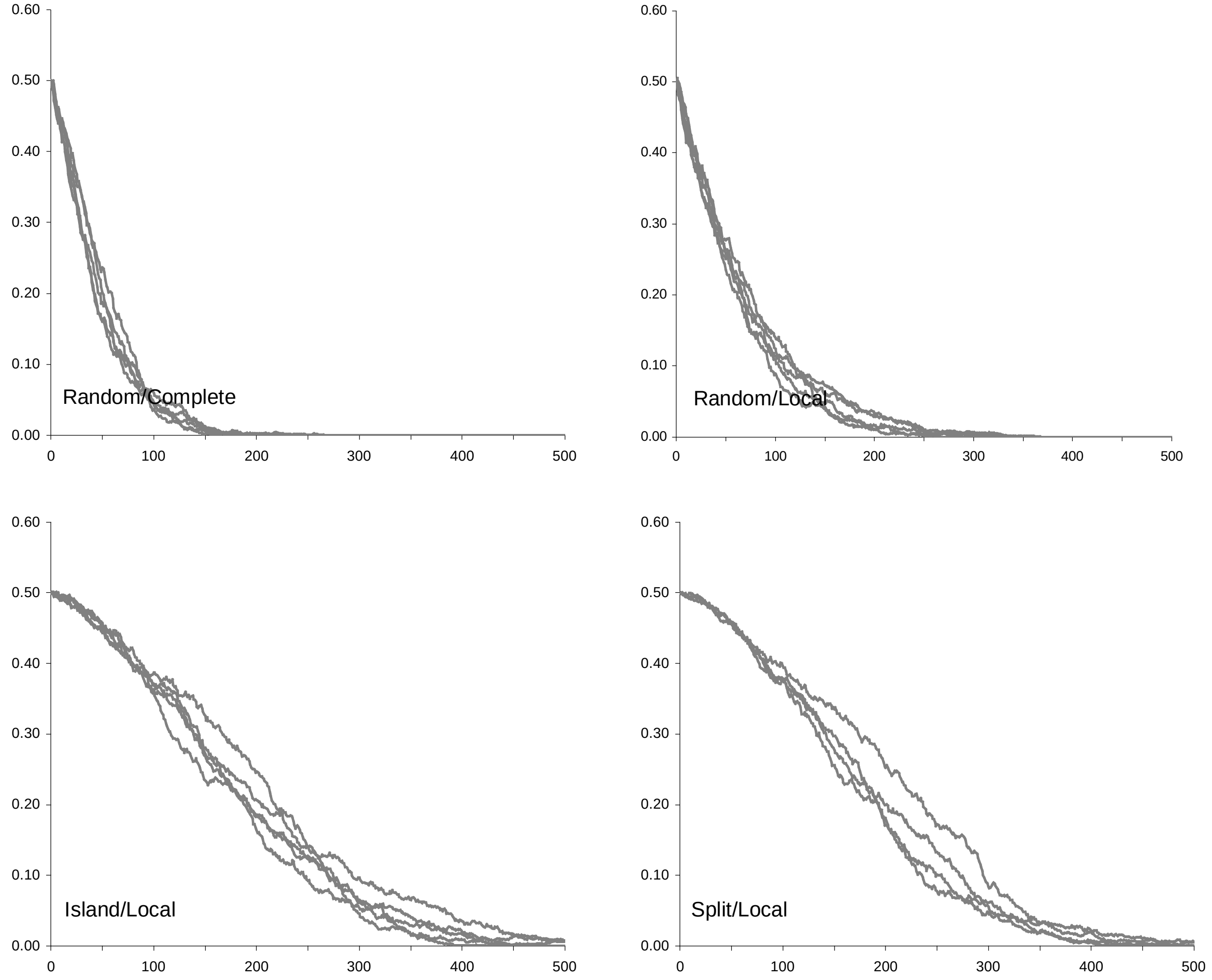

Figure 8: Graphs of α_t over 500 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = 0.125$, $b = 0.1$, and $K = 128$. Notice the significant increase in long-term viability with the introduction of the local network connections, and the additional positive effects conferred by the island and split distributions. Also, note the infamous S-shaped curve of historical linguistics in the island and split distribution plots.

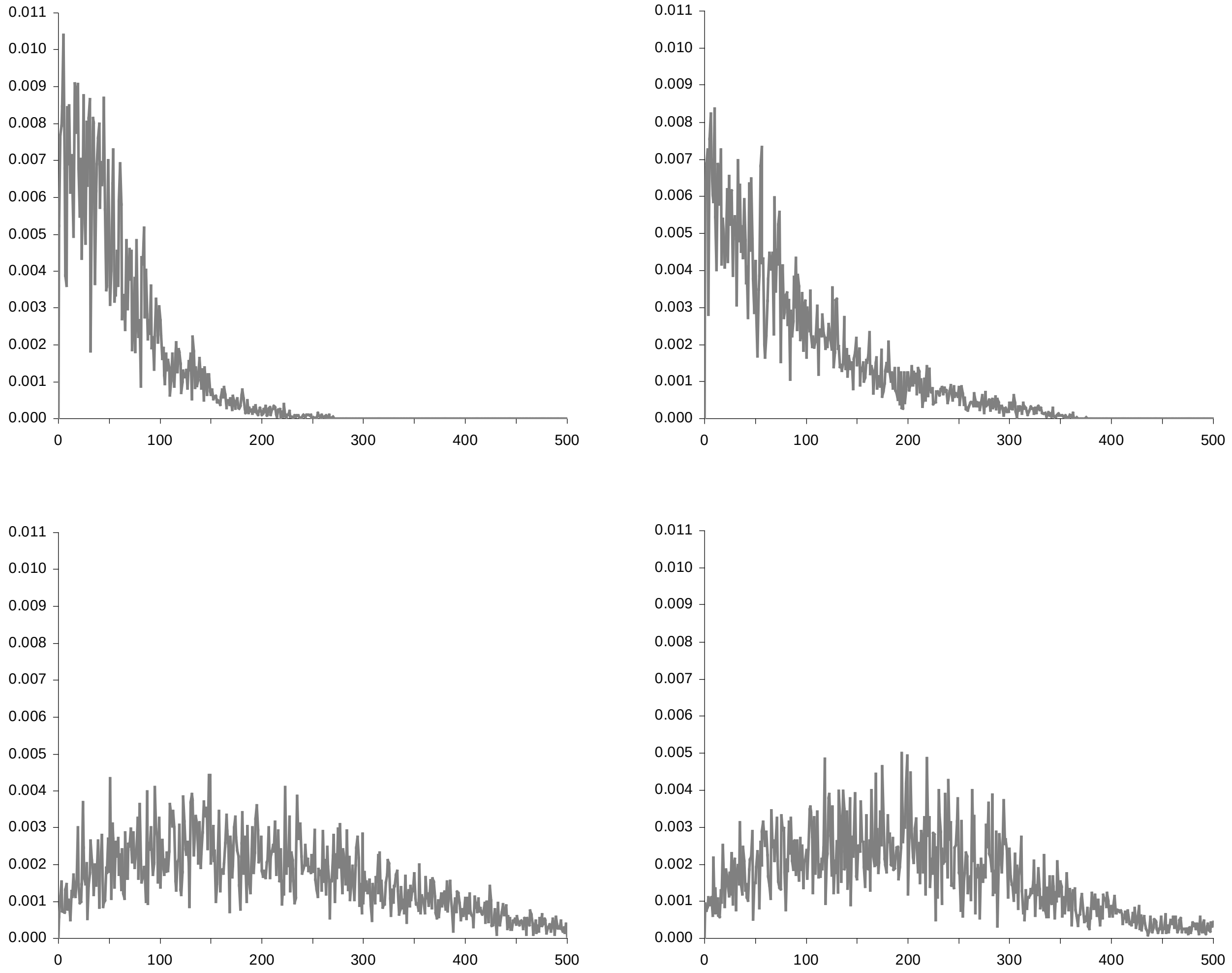

Figure 9: The average generational change of α_t, $\overline{\Delta\alpha}$, over 500 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = 0.125$, $b = 0.1$, and $K = 128$. Notice that the magnitude and variability is initially greatest in the random/complete system, with each decreasing as α approaches 0. Also, note the initial magnitudes and trajectories of the island and split distributions: telltale signs of dense clustering and edge effects.

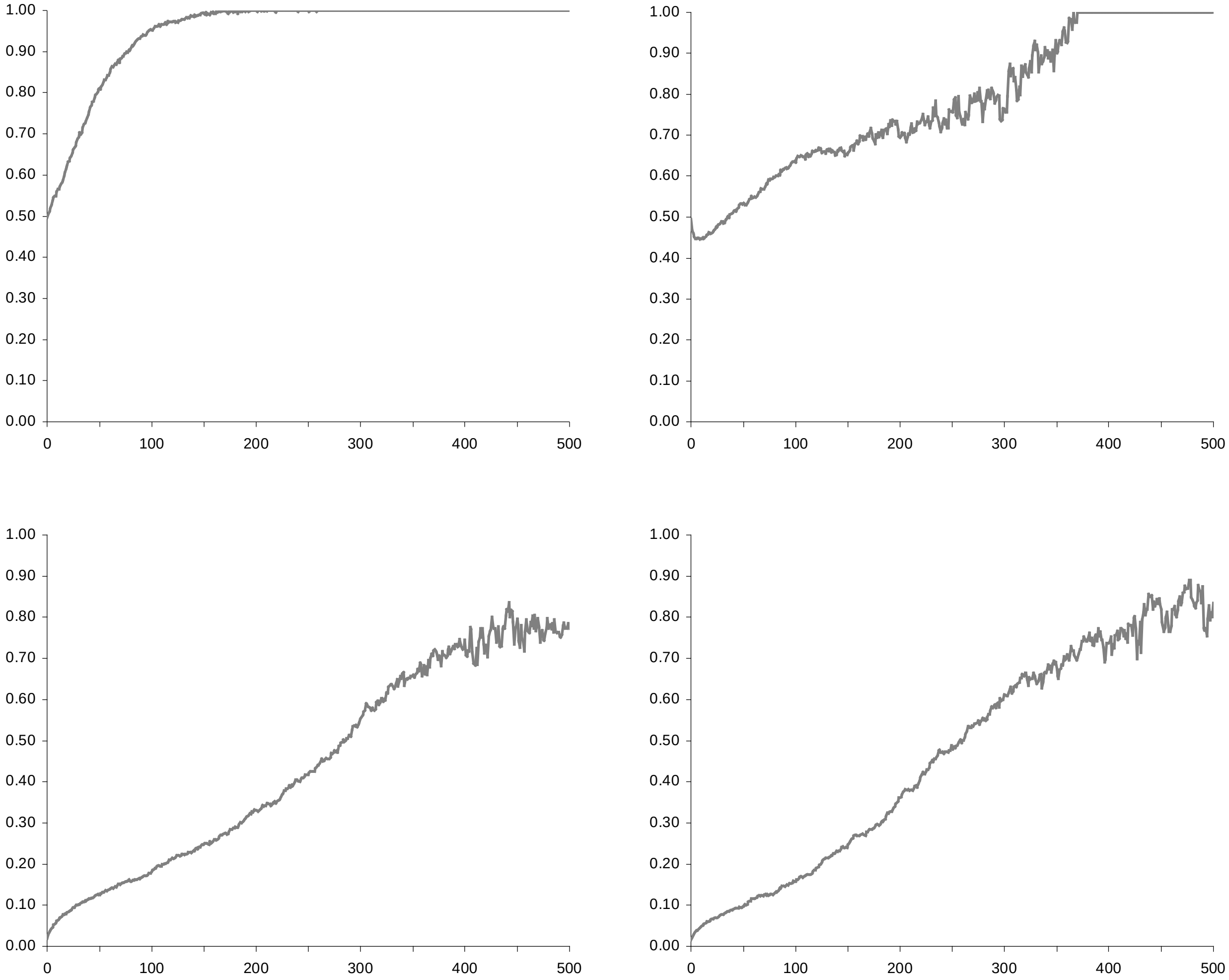

Figure 10: The average local variance over 500 generations in a population of 10,000 individuals using the TLA algorithm, with $\alpha_0 = 0.5$, $a = 0.125$, $b = 0.1$, and $K = 128$. Notice that the local variance rapidly approaches 1 in the random/complete system, there is an initial decrease and subsequent quasi-linear growth in the random/local population, and that both the island/local and split/local systems start at extremely low levels and then increase almost linearly, similar to the random/local system.

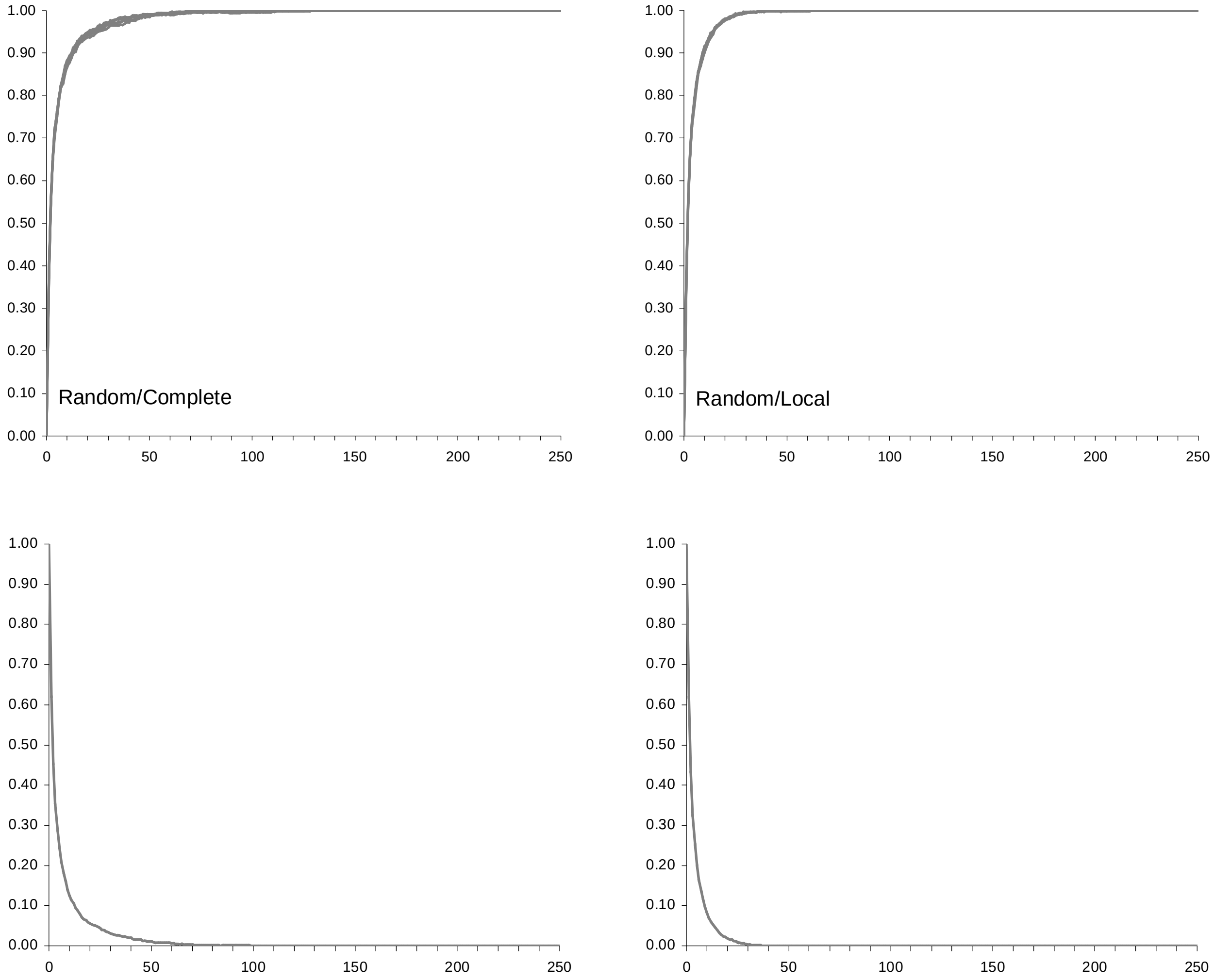

Figure 11: Graphs of α_t (top) and the average local variance (bottom) over 250 generations in a population of 10,000 individuals using the Batch-Error algorithm, with $\alpha_0 = 0$, $a = 1.0$, $b = 0.9925$, and $K = 128$. Notice that the locally connected population reaches the stable fixed point $\alpha_t = 1$ quicker than the completely connected one, and that local variance decreases faster in the locally connected system.

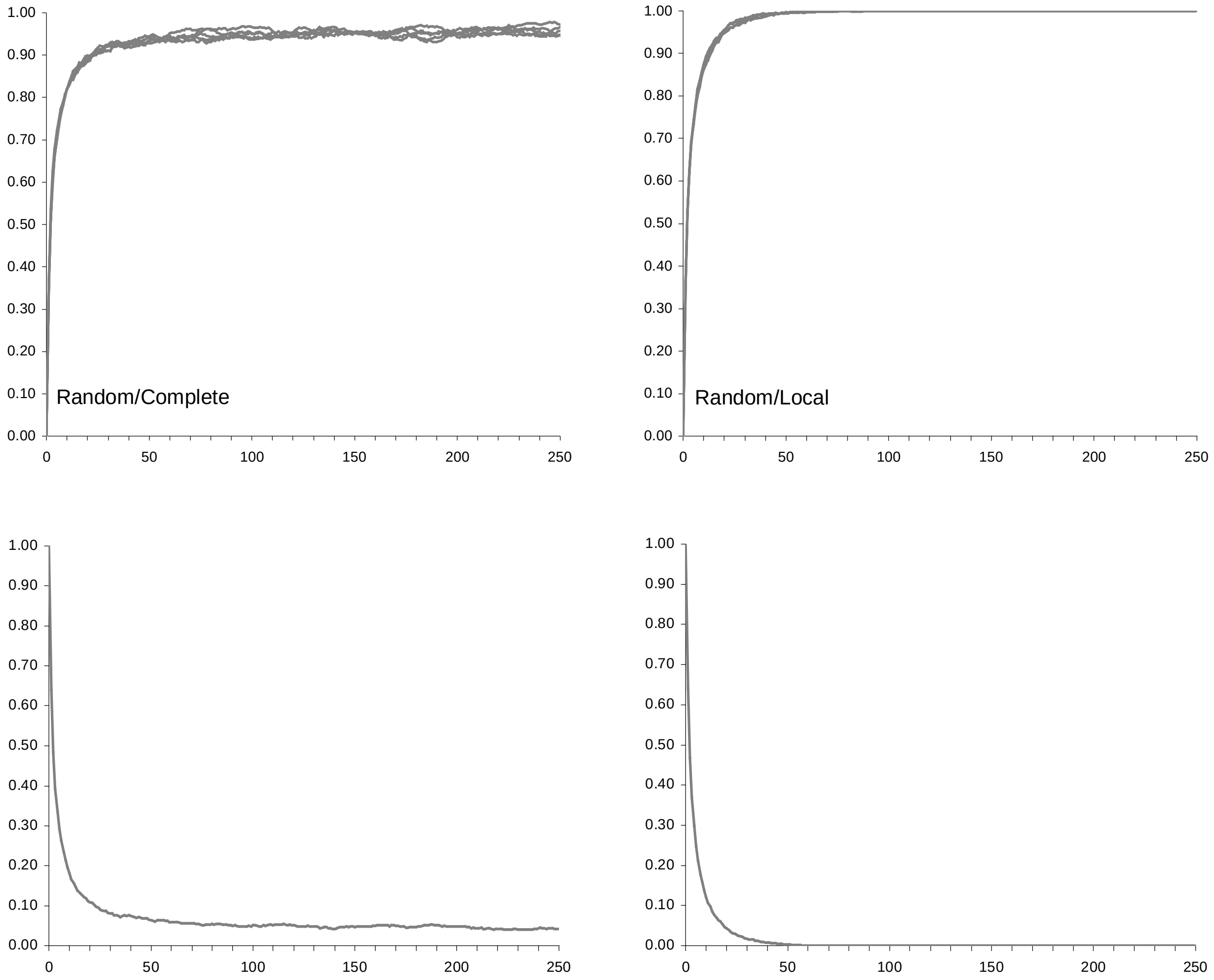

Figure 12: Graphs of α_t (top) and the average local variance (bottom) over 250 generations in a population of 10,000 individuals using the Batch-Error algorithm, with $\alpha_0 = 0$, $a = 1.0$, $b = 0.992$, and $K = 128$. Notice that the completely connected population goes to the stable fixed point $\alpha_t < 1$ while the locally connected system surpasses the fixed point and goes to $\alpha_t = 1$. Since the maturation time is relatively high and fixed probability used to decide ties in number of sentences equals 1, the system remains at $\alpha_t = 1$ rather than decreasing towards the stable fixed point.

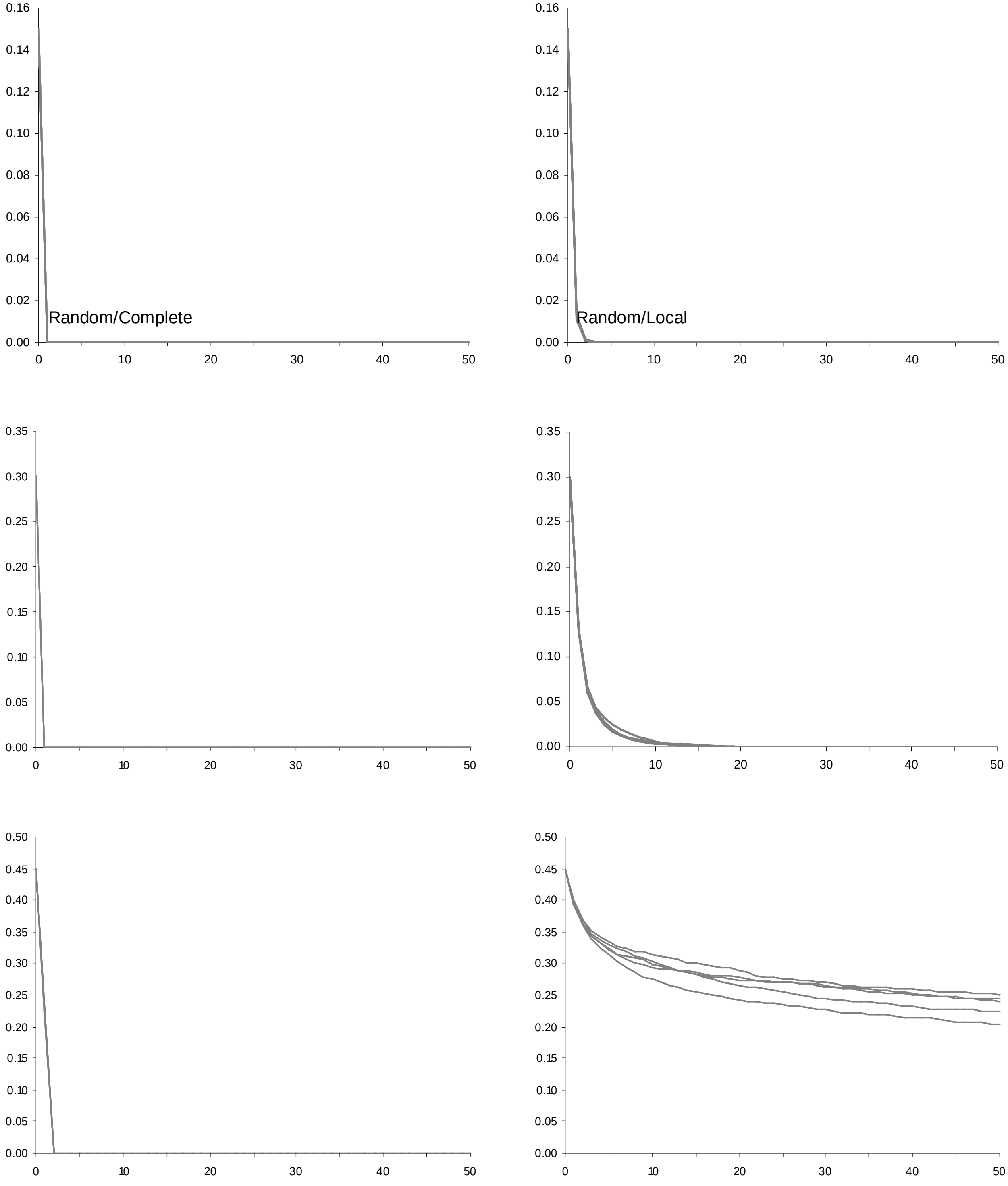

Figure 13: Graphs of α_t over 50 generations in a population of 10,000 individuals using the Batch-Error algorithm, with $a = b = 0.5$, $K = 128$ and $\alpha_0 = 0.15$ (top), $\alpha_0 = 0.3$ (middle) and $\alpha_0 = 0.45$ (bottom). Notice the effects of local network-induced clustering, with quantitative outcomes eventually diverging radically as α_0 approaches $\alpha_* = \frac{1}{2}$.

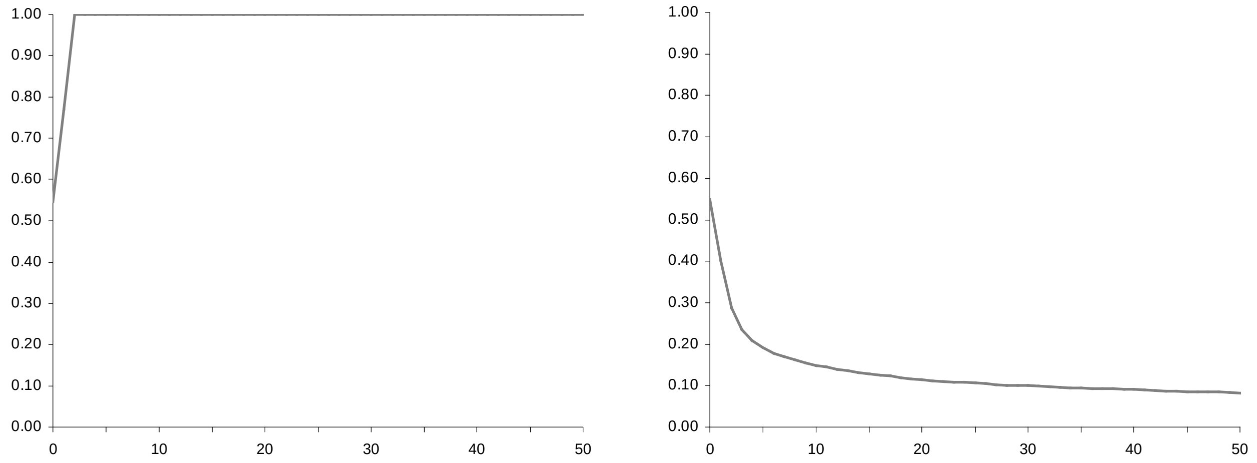

Figure 14: Graphs of the average local variance over 50 generations in a population of 10,000 individuals using the Batch-Error algorithm, with $a = b = 0.5$, $K = 128$ and $\alpha_0 = 0.45$. Notice the marked difference between the completely and locally connected populations, with the locally connected system exhibiting an extremely high level of clustering within the first few generations, ultimately leading to stability.

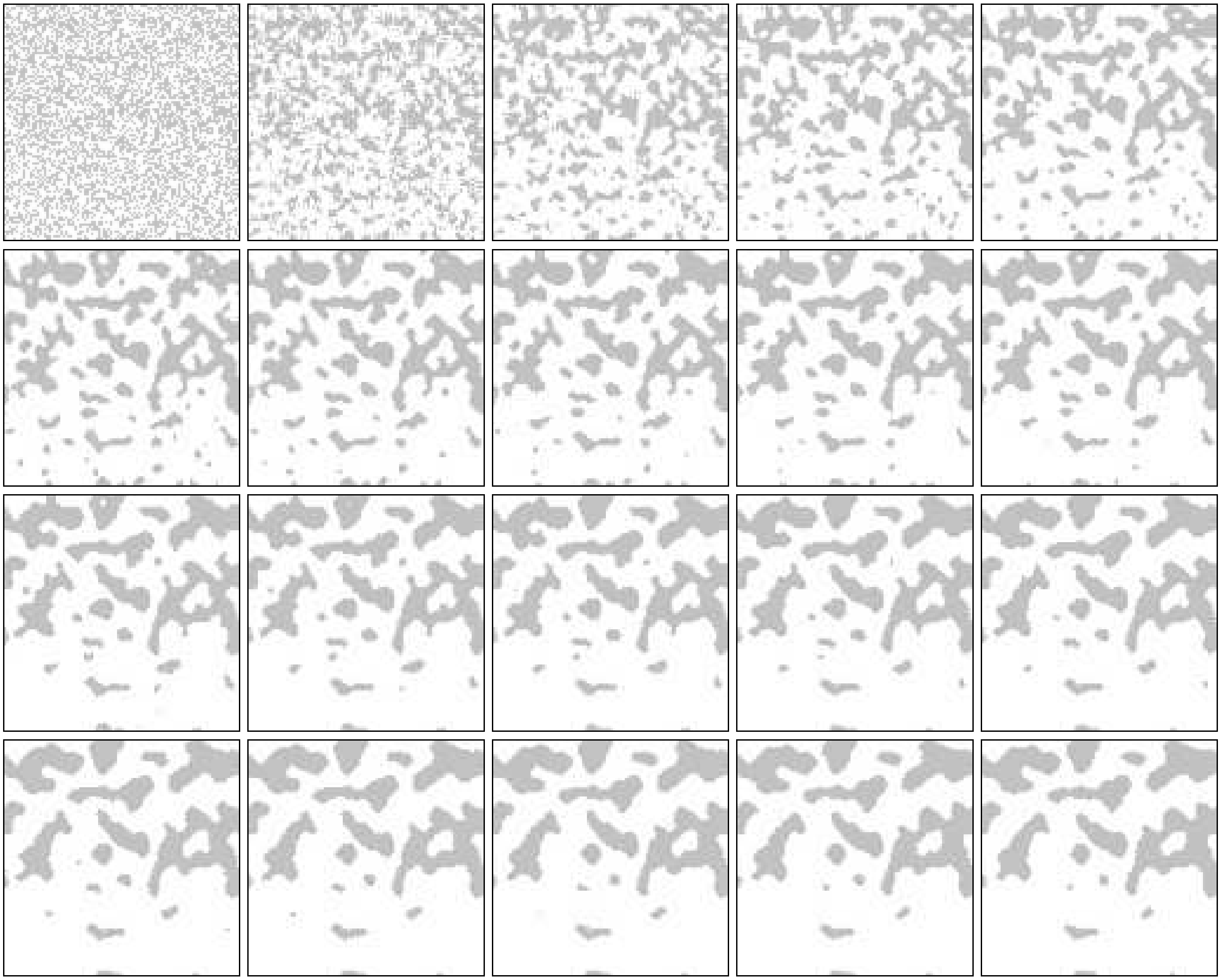

Figure 15: ALingua screenshots of the first 20 generations of a randomly distributed population of 10,000 individuals using the Batch-Error algorithm, with $\alpha_0 = 0.45$, $a = b = 0.5$, and $K = 128$. Notice the rapid clustering and relative stability induced by the local network connections.

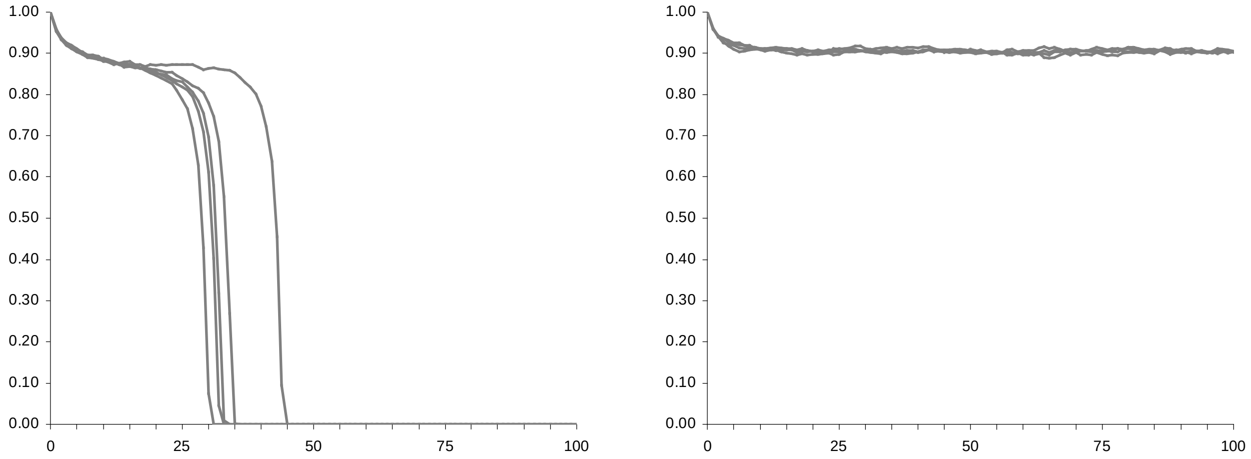

Figure 16: Graphs of α_t over 100 generations in a population of 10,000 completely connected individuals using the Cue-Based algorithm, with $p = 0.15$ (left), $p = 0.151$ (right), $\tau = 0.1$, and $K = 128$. Notice how a small increase in p has given rise to the development of an additional stable fixed point.

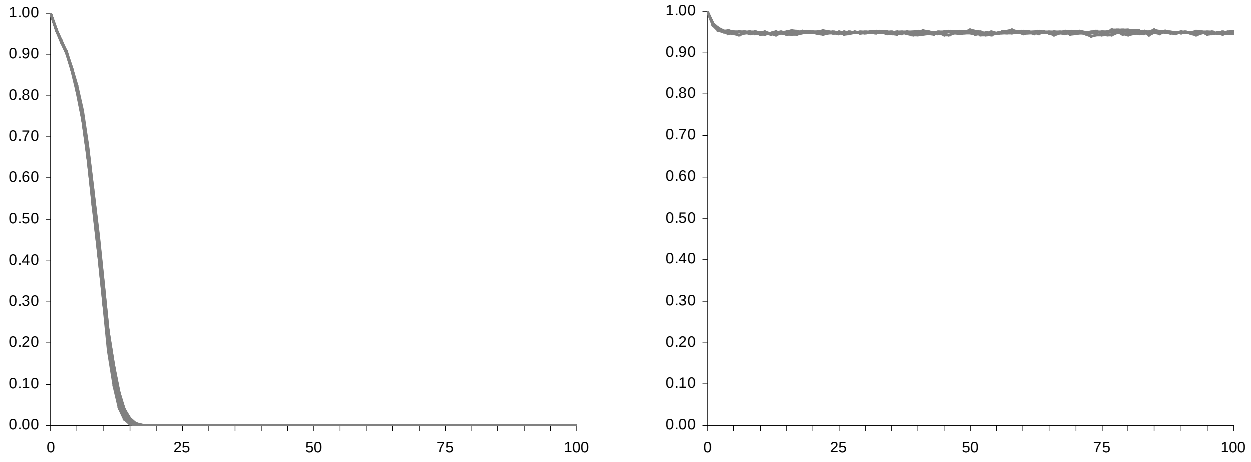

Figure 17: Graphs of α_t over 100 generations in a population of 10,000 locally connected individuals using the Cue-Based algorithm, with $p = 0.151$ (left), $p = 0.155$ (right), $\tau = 0.1$, and $K = 128$. Notice how the graph of $p = 0.151$ differs from that in the completely connected population (Figure 16, right). In particular, the bifurcation point has not been reached. However, an increase of 0.004 percent yields the expected additional stable fixed point.

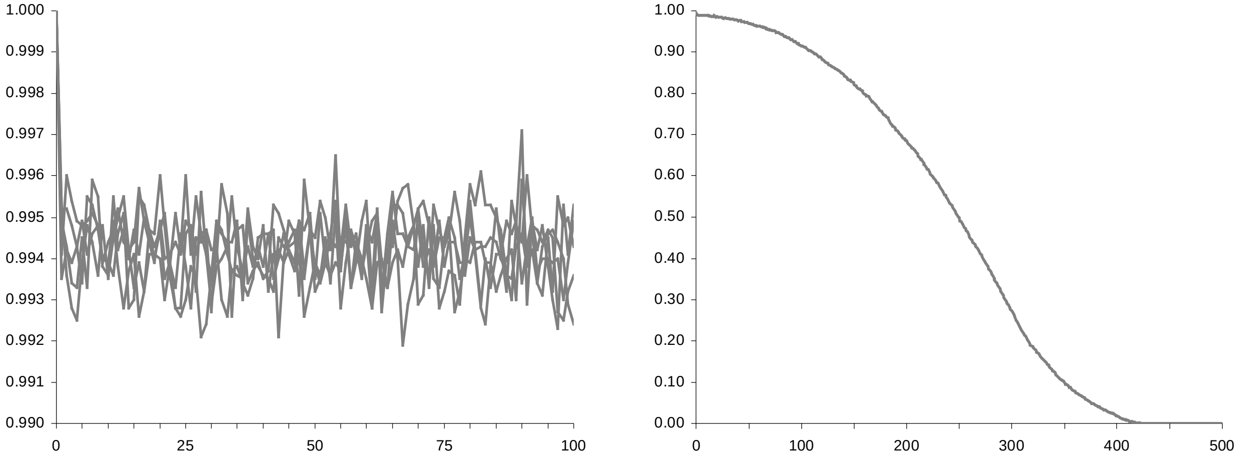

Figure 18: Graph of α_t over 100 generations in a population of 10,000 randomly distributed, locally connected individuals using the Cue-Based algorithm, with $p = 0.18$, $\tau = 0.1$, and $K = 128$ (left), and α_t over 500 generations in a population of 10,000 locally connected individuals using the Cue-Based algorithm, with $\alpha_0 = 0.995$, $p = 0.18$, $\tau = 0.1$, and $K = 128$ (right). The plot on the left shows that an initial population size $\alpha = 0.995$ should be stable, however the graph on the right shows that if the system is seeded with an island of non-cue-driven speakers the stability of the system is greatly effected. Also note the appearance of another S-shaped curve.

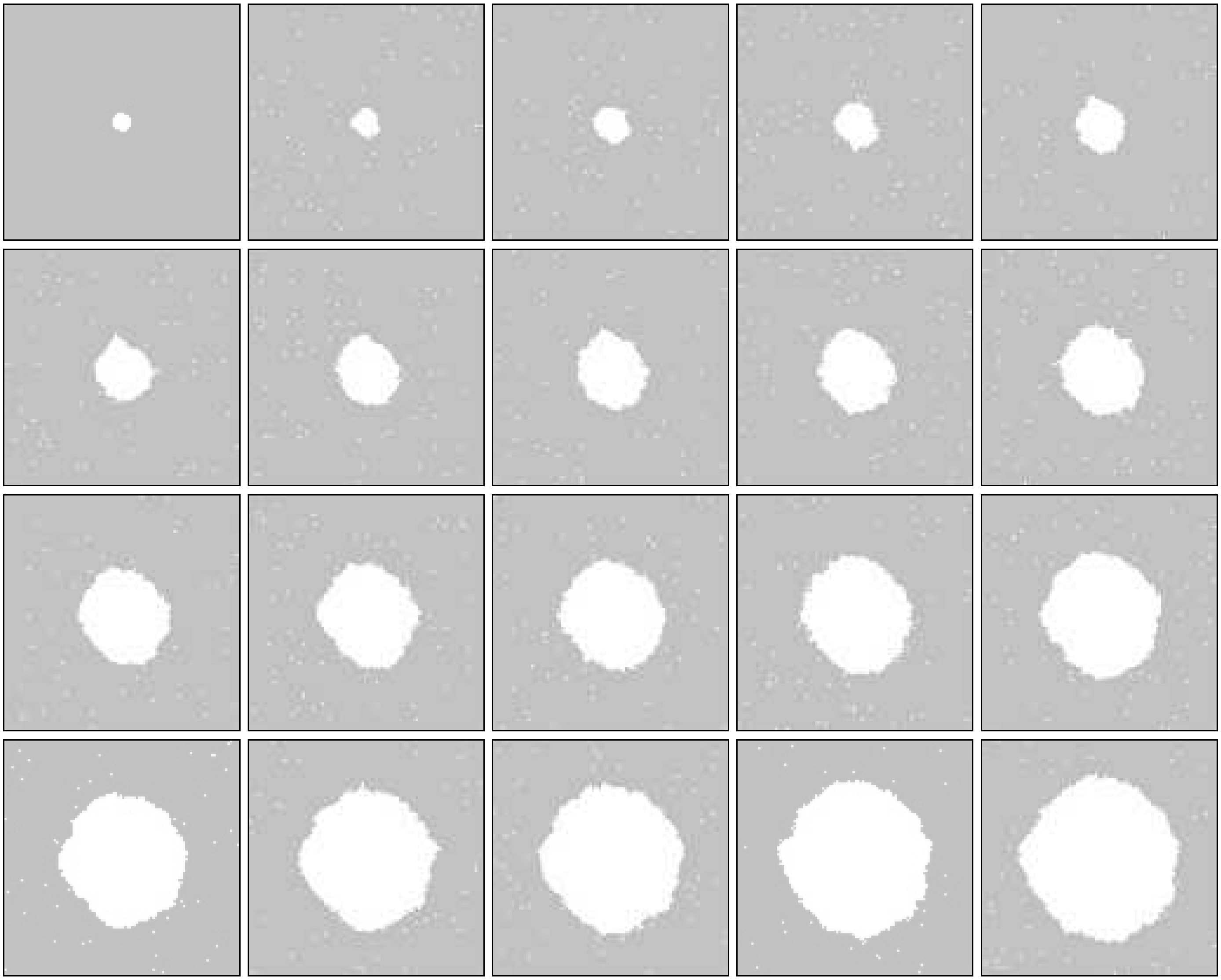

Figure 19: ALingua screenshots, taken every 10 generations, of the first 200 generations of a population of 10,000 locally connected individuals using the Cue-Based algorithm, with $\alpha_0 = 0.995$, $p = 0.18$, $\tau = 0.1$, and $K = 128$, and with the initial 0.005 of the non-cue-driven population distributed as an island cluster. Notice how the initial cluster expands over time. Eventually, the system converges to the stable fixed point $\alpha_t = 0$.