Effects of Cooperative and Competitive Coevolution on Complexity in a Linguistic Prediction Game

Nick Moran\(^1\) and Jordan Pollack\(^1\)

\(^1\)DEMO Lab, Brandeis University, 415 South St, Waltham, MA
eemeteryax@brandeis.edu

Abstract

We propose a linguistic prediction game with competitive and cooperative variants, and a model of game players based on finite state automata. We present a complexity metric for these automata, and study the coevolutionary dynamics of complexity growth in a variety of multi-species simulations. We present quantitative results using this complexity metric and analyze the causes of varying rates of complexity growth across different types of interactions. We find that while both purely competitive and purely cooperative coevolution are able to drive complexity growth above the rate of genetic drift, mixed systems with both competitive and cooperative interactions achieve significantly higher evolved complexity.

Introduction

Explaining the causes and mechanisms of the evolution of increasing complexity in biological systems, and reproducing those mechanisms in silico remains an open problem for the field of artificial life (Bedau et al., 2000; Taylor et al., 2016). A variety of hypotheses have been proposed in both the biological literature at large, and within the evolutionary computation community. In his book Full House (Gould, 2011), biologist Stephen J. Gould suggests that the apparent increase in biological complexity over evolutionary time is explained by the process of random mutation performing a drunkard’s walk through the space of possible phenotypes. Because there is a lower bound of complexity at zero, this random walk will, over time, reach higher and higher maxima. On the other hand, Richard Dawkins has offered co-evolutionary “arms races” to explain the development of ever-higher complexity (Dawkins and Krebs, 1979). In this hypothesis, co-evolutionary competition drives a positive feedback cycle of complexity growth, in which each new innovation by one competitor drives selection for further innovation in the other.

There are a variety of challenges when examining this sort of question, but perhaps chief among them is the challenge of satisfactorily defining complexity. The literature contains a wide variety of definitions in use across computer science (Wolfram et al., 1984; Adami, 1998; Dorin and Korb, 2010), biology (Adami et al., 2000; Dawkins, 2010), and in other fields. A common thread in many of these definitions (though certainly not all), is the notion of incompressibility - that is, that things are complex when they cannot be succinctly described. In particular, the concept of algorithmic complexity developed by Kolmogorov and Chaitin defines the complexity of a string as the length of the shortest program on a universal Turing machine which generates that string (Kolmogorov, 1968; Rissanen, 1978). Unfortunately, this measure is difficult to use in practice, as it cannot be computed in the general case. In the field of automata theory, the state complexity of a regular language is defined as the size of the smallest finite automaton which can recognize that language (Yu, 2001). This measure is conceptually similar to that of Kolmogorov and Chaitin, but can be calculated by a straightforward algorithm. We will present a variant of this metric, and use it to evaluate the complexity our evolved phenotypes.

In this work, we propose a simple model to serve as a testbed for these and other hypotheses. This model consists of a co-evolutionary game between multiple species, a representation of strategies for this game, and a complexity measure over that representation. The game is a two-player, iterated game in which each individual generates a binary string, and receive fitness points depending on whether their string matches or differs from their opponent’s string at each index. This game has both a competitive form, in which the two players have opposing goals (that is, one seeks to match and the other seeks to differ), and a cooperative form, in which the two players share a goal (that is, both seek to match, or both seek to differ). We model players of this game as finite state machines, which have been extended to include multi-player interactions. We then calculate the state complexity of a strategy by applying a minimization algorithm to find the smallest finite automaton which can generate that strategy. Further details on each of these component can be found in their corresponding sections.

Using this model, we can construct a variety of ecological networks, in which we define either competitive or cooperative pair-wise relationships between multiple species. In networks with more than two species, a single species may
have multiple such relationships, some competitive and others cooperative, and will derive fitness from all of them. We then compare the growth of complexity of various network topologies in order to determine which, if any, promote complexity, and to which degree.

**Model**

**Linguistic Prediction Game**

We present a simple prediction game which we term the Linguistic Prediction Game. In this game, two players iteratively generate symbols (in this work, 0 or 1) and are rewarded points each round depending on their symbol and the symbol of the other player. Symbol generation in each round is done simultaneously - neither player may know what symbol the other has chosen to produce. However, players may condition their next symbol on the history of symbols played so far by both players in the current game.

Many different payoff matrices can be defined, but we concern ourselves with a few simple matrices which model cooperation and competition. In these games, each player either seeks to produce a symbol which matches that of the other (in other words, both produce zero or both produce one), or seeks to produce a symbol which mismatches that of their opponent (in other words, one player produces zero and the other produces one)\(^1\). If the goals of both players are aligned, meaning they both wish to match or both wish to mismatch, then we say the game is cooperative, and if their goals are opposed, then we say the game is competitive.

In each round, a player receives one point if their goal (match or mismatch) succeeds, and zero points otherwise. In this work, we consider a game which is played over an infinite number of rounds (similar to the formulation of an infinite iterated prisoner’s dilemma in (Lindgren, 1992)), and we define the score of a particular player via a limit:

\[
S(p) = \lim_{r \to \infty} \frac{\sum_{i=0}^{r} r \cdot \text{reward}(p, i)}{r}
\]

Where \(S(p)\) indicates the total score of player \(p\), \(r\) the number of rounds played, and \(\text{reward}(p, i)\) indicates the reward received by player \(p\) in round \(i\).

We consider two specific types of game matrices, as shown in Figure 1, which we term COOP and COMP. COOP is a cooperative game in which both players receive a point if they output the same symbol. In this game, players must learn to coordinate their outputs to receive a maximal score. Players seek to establish a shared convention (a language) to which they can both adhere. COMP, on the other hand, is a competitive game in which one player receives a point if they output the same symbol, and the other receives a point if they output different symbols. Here both players must try to predict the pattern of their opponent while simultaneously preventing prediction of their own pattern.

Note that there are two additional isomorphic formulations of each game - COOP also occurs when both players seek to output different symbols, and COMP also occurs when the roles of the two players are swapped. In our experiments, the choice of which form of each game to use has had no effect on the results.

**Multi-game Ecosystems**

We extend this linguistic prediction game model by embedding multiple instances of it in a network of interacting species, which we term an ecosystem. An ecosystem consists of a set of species, and a set of pairwise connections between them. Each connection is labelled with a game matrix which governs the interaction between members of those two species. In this way, a single organism may interact with other organisms of multiple different species, and will potentially play a different version of the game with each of those species.

We examine several of these ecosystem models here, as shown in Figure 2. The first is a trivial baseline model with a single species and no games being played. This serves as a control datapoint which allows us to understand the complexity which tends to arise only from a random walk through mutations. The next two models are a pair of two-species models which are fully cooperative (playing the COOP game), and fully competitive (playing the COMP game). These two models allow us to examine the behavior that occurs when the players have either aligned or opposing goals. We also consider an ecosystem which features a mix of cooperative and competitive games, which we term “Three-Species Mixed”. This ecosystem consists of three species, \(A\), \(B\) and \(C\). \(A\) plays the COOP game against \(B\)
and the COMP game against C, and there is no direct interaction between B and C. This system is inspired by the
three-player pursuit/evasion game introduced by Ficici (Ficici et al., 1998). For comparison with this three-species
system, we also examine a purely competitive three-species system in which A competes with B and with C, but with
opposite goals - that is, A attempts to match B but mismatch C. We term this system “Three-Species Comp”. Finally, we
consider a four-species system extension of Ficici’s game in which both species A and B compete, and each has a coop-
erative partner, C and D respectively. A and C play COOP as do B and D. We term this system “Four-Species Mixed”.

Critically, in ecosystems in which one individual may interact with multiple other species, there is no signal or indi-
cation of which such other species is being interacted with. For example, in the Three-Species Mixed system, individu-
als of species A will not know ahead of time whether they are interacting with a member of species B, and therefore
playing COMP, or a member of species C, and therefore playing COOP. To the extent that a member of species A
exhibits different behavior in these two situations, it must do so by learning to recognize the species of its opponent
based on its opponent’s outputs, and adjust accordingly.

Organism Model

Individual organisms are represented as finite state machines in which each state is labelled with a symbol to be produced
either zero or one, and each state has an outgoing transition link for each symbol. State transitions occur according to the
symbols produced by the other player in game. Additionally, each finite state machine has a special initial state in which it
will start at the beginning of each game. Thus, an organism can be represented as a tuple $S, T, L, s_0$, where $S$ is a set
of states, $T : S \times \{0, 1\} \rightarrow S$ defines state transitions for each state in $S$ for both 0 and 1, $L : S \rightarrow \{0, 1\}$ defines the
output symbol for each state in S, and $s_0$ indicates the initial state.

To simulate a game, each player is first set to its initial state $s_0 \in S$. Then, each player produces the symbol indicated
by the label on its current state, and performs the state transition indicated by the symbol produced by the oppo-

tent. To calculate the score received by each player, this process is repeated until the two players enter a loop (note
that players are fully deterministic, and possess a finite number of internal states, and therefore a pair of players must
enter a finite-length loop after a finite number of rounds). To detect loops, we track the pairs of states of each player
in each round, and wait for the system to return to the same pair of states as it has been in before. Each player’s score for
the game is then its average score over the loop. Note that this means interactions before the loop is entered (i.e. the
transient state of the system prior to its stable cycle) do not contribute to the score, as they contribute an infinitesimal
amount in the limit of an infinite game.

As an example, consider the two organisms pictured in

Figure 3. Let the first organism be A, with states $A_0$ and
A1, and the second be B with states $B_0$ and $B_1$. A game
between these two players will begin in states $\{A_0, B_0\}$, and
will proceed as $\{A_0, B_0\} \rightarrow \{A_1, B_0\} \rightarrow \{A_1, B_1\} \rightarrow \{A_0, B_1\} \rightarrow \{A_1, B_0\}$. At this point, the game has entered
a loop, and the score can be calculated using a game matrix.

Mutation We further define a set of mutation operators
which alter the structure of an organism$^2$. These are:

- Add a new state.
- Remove a state.
- Redirect a transition link.
- Change the symbol produced in a state.

$^2$These mutation operators are similar to the class of structural
mutators used in (Angeline et al., 1994) and other network-based
evolutionary algorithms.
Adding a new state results in a state with randomly-assigned outgoing transitions (possibly pointing to itself), and with a random output symbol. Note that this new state will not yet be the target of any external transitions - it must later be linked by redirection mutation. Removing a state requires randomly selecting new destinations for every link which previously pointed to the deleted node. Further, if the deleted node was the initial state, a new initial state is randomly selected from the remaining nodes. If the organism has only one state, a deletion mutation instead has no effect. A redirection mutation simply selects a random transition link and selects for it a new destination. This may result in making some nodes unreachable, or making some previously unreachable nodes reachable. Changing the symbol produced by a state is self-explanatory.

**Evolutionary Simulation**

An evolutionary simulation consists of the following steps. First, for each species in an ecosystem, a population of organisms is initialized with minimal finite state machines - a single state with a random label, and transition links leading back to that single state. We then simulate a series of generations. Each generation consists of an evaluation period, in which individuals interact according to the games specified by the ecosystem to accumulate fitness, and a reproduction period in which individuals produce offspring which form the populations of the next generation.

In the evaluation period, for each pair of interacting species, an all-vs.-all set of games are played, as shown in Figure 4. That is, if species $A$ plays the COOP game with species $B$, each member of $A$ will play that game with each member of $B$. For each organism, its total fitness across all interactions is tallied.

In the reproduction stage, for each species, a new population is generated using fitness-proportional selection. That is, a number of offspring are created equal to the population size of each species where the parent of each offspring is chosen at random from the current population of that species, with probability proportional to each individual’s fitness accumulated during the evaluation period. Offspring are then subjected to mutation as described previously. In this work, we use a constant mutation rate of one mutation per offspring, with all four types being equally likely. Once the new population has been generated, the previous generation is discarded.

**Complexity Metric**

Our goal in this work is to measure the growth of complexity of organisms over the course of an evolutionary simulation. To do so, we require a quantitative metric of complexity. A simple approach would simply be to count the number of nodes and transitions in a particular organism, and define those with more nodes to be more complex. However, consider the example of an organism with a great many nodes, all of which are labelled to produce the symbol 0. This simple metric would label this organism as very complex, but its behavior would be identical to an organism with a single node! This is clearly unsatisfactory. In order for an organism with a large network to be truly complex, it seems that it should be the case that all of its nodes are essential to its strategy.

We take inspiration from the concept of Kolmogorov complexity, which (informally) defines the complexity of a string as the length of the shortest computer program which generates that string. We define the complexity of an organism as the size of the smallest organism which produces an identical strategy. Thus, the large organism in the example above would have a very small complexity, as its strategy can be produced by an organism with a single node. Similarly, unreachable nodes will not contribute to the complexity of an organism, as those nodes could be removed without altering the strategy.

To calculate the minimal organism which produces the same strategy, we turn to automata theory. A slight modification of Hopcroft’s algorithm for minimization of deterministic finite state machines (Hopcroft, 1971) allows us to efficiently compute the smallest equivalent organism. As in the base algorithm, we begin by removing all disconnected nodes. We then apply the partition-refinement process, but instead of starting with an initial partition of accepting and
non-accepting states as in Hopcroft’s algorithm, we begin
with a partition of nodes by output label. From there, a
straightforward application of the standard algorithm pro-
duces a minimal equivalent machine. The total number of
nodes in this minimized machine is then used as our com-
plexity metric for that organism.

Experiments & Results
For each of the five ecosystems described above, we per-
form 20 experiments of 10,000 generations each, with each
species having a population size of 50. For each genera-
tion, the complexity (the size of the minimal equivalent ma-
chine) of each organism is computed, and we record the me-
dian complexity for each population. We use the median
to avoid giving undue influence to a single complex mutant
which does not manage to spread its complexity through-
out the population. We then take the mean of these medians
over the 20 experiments to determine the characteristic tra-
jectory of complexity over time for each ecosystem. To ana-
lyze the trajectory of complexity growth within each species
at the end of the simulation, we compute a line of best fit
over the complexity values for the last 2,500 generations of
each experiment using the least squares method, and report
the slope of this line. Species which sustained complexity
growth through the end of the experiment display a posi-
tive slope, while those whose complexity growth had slowed
show a near-zero slope. We will present results based on this
trajectory, as well as the final level of complexity at the end
of 10,000 generations. Figure 5 shows the trajectories of
average complexity for each ecosystem. Figure 5 summa-
rizes final and maximal average complexity values as well
as complexity trendline slopes for each ecosystem.

Single-Species (Control)
In this ecosystem, we have only a single species which ex-
periences no selective pressure - all organisms are equally
likely to reproduce. Thus, the only driver of increased com-
plexity is the random walk of genetic drift. Because this
walk is bounded below at 1, we should expect a slight up-
ward trend over time. This is indeed what we observe, with
the populations acheiving an average median complexity of
about 12 nodes after 10,000 generations, with a high-water
mark of 14.24 reached after 7,589 generations. By the end
of the simulation, the control species had reached a near-zero
rate of complexity growth. This will serve as a baseline for
comparison to other systems - complexity growth above this
rate should be the result of selective pressure.

Two-Species Cooperative
This ecosystem shows the behavior of two species with a
purely cooperative interaction. Individuals which are able to
consistently match the output of the other species are favored
for reproduction. We find that this system is more amenable
to complexity growth than the control - both in terms of its
final average complexity as well as the rate of growth. The
two species reach final complexities of about 18 nodes after
10,000 generations, with highs of 19.78 and 22.52 reached
after 6,388 and 8,507 generations respectively. By the end
of the simulation, both species showed sustained, although
slow, complexity growth at a rate of around 1.5 additional
nodes per 1,000 generations. In some sense, this growth of
complexity is surprising - the fully cooperative game can be
played perfectly by two organisms with a single state, so it
seems that extra states would be superfluous. From manual
inspection of organism controllers, we observe that there is
pressure towards strategies which are tolerant against muta-
tions of their opponents. For example, consider a population
of single-node organisms which all output 0. In each gen-
eration, there will likely be some offspring that have under-
gone a label-flipping mutation and instead output 1. Interac-
tion with these mutants would result in zero fitness for the
all-0 players. However, a two-state organism which always
copies the last move of its opponent would achieve maxi-
mum fitness against both the original all-0 players and the
all-1 mutants. Similar interactions drive further complexity
development.

Two-Species Competitive
This ecosystem is the mirror of the cooperative environment
above - the two species compete to achieve opposing goals:
one receives fitness for matching output symbols, and the
other for mismatching. Here we observe a very fast ini-
tial growth of complexity, with species reaching an average
complexity of 10 nodes after only about 400 generations.
By comparison, the purely cooperative system took 1,400
generations to reach this level, and the control took over
3,200 generations. However, this fast initial growth is not
sustained. The system reaches final complexity values of
about 16 nodes, with highs of 19.88 and 20.7, all slightly
lower than those acheived by the cooperative model. The fi-
nal complexity trendlines show a modestly increasing trajec-
tory for one species (0.5 additional nodes per 1,000 gener-
ations) and a modestly decreasing trajectory for the other (~
0.8 nodes per 1,000 generations). This system appears to fall
victim to a form of mediocre stable state as described by (Fi-
cici and Pollack, 1998), in which the system ceases to make
progress as measured by an external metric (in our case,
complexity). Instead, the species are driven into a cyclical
pattern of what Ficici terms “convention-chasing”. That is,
species spend all of their available mutational change try-
ing to keep up with the changes of their opponents, rather
than become more complex. In particular, selection seems to
favor cycles of label-changing mutations, which pushes
the populations through variants of the same core strategies
but with different output symbols.
Three-Species Mixed

This ecosystem contains three species with both cooperative and competitive interactions. Unlike the previously discussed systems, the roles of the species are not symmetric. Species A faces both cooperative and competitive interactions, species B faces only a competitive interaction - its interaction is locally equivalent to those in the two-species competitive model, and species C faces only a cooperative interaction - locally equivalent to those in the two-species cooperative model. We observe significantly higher complexity values in this system than found in any of the two-species models. Species A reaches a final complexity of about 26 nodes, with a high of 29.12. Species B reaches a final complexity of about 20 nodes, with a high of 23.14, and species C reaches a final complexity of about 41 nodes, with a high of 41.72. On the other hand, the final complexity trajectories show that growth has slowed to less than one additional node per 1,000 generations for all three species by the end of the simulation. Species C in particular appears to suffer a period of regression around the 8,000th generation that arrests its otherwise consistent complexity growth. The exact cause of this regression is unclear.

The complexity values for B and C are both higher than those found in the two-species competitive and cooperative games respectively, although the values for B only slightly so. The bulk of the increase in complexity seems to occur from the interaction between A and C, with both significantly exceeding the complexity levels observed in the purely cooperative game. The mechanism driving this growth seems to be similar to that in the cooperative game - C is driven to develop tolerance against changes in the behavior of A. However, the changes in the behavior of A are no longer merely the result of genetic drift, they’re driven by selection pressure exerted by B.

Three-Species Comp

It might fairly be asked whether the increase in complexity shown in the previous system was merely due to the introduction of a third species - perhaps more players results in more complexity. To test this hypothesis, we consider a system with the same interactions, except that the game between A and C has changed to COMP. Here we find no in-
increase in complexity above that observed in the two-player case. In fact, the convergence to mediocre stability is even stronger, with all three species quickly reaching a point of stagnant complexity. The three species reach final complexities of 13.7, 14.0 and 13.0 respectively. They reach highs of 16.1, 16.5 and 16.1, respectively. All of these values are slightly below those observed in the two-species competitive case. We also see that species mixed system, and well below the levels observed in the three-species mixed system. Further, all three species display final trajectory trendlines with slopes very close to zero. We therefore conclude that the interaction of competition and cooperation is driving the increase in complexity, not merely the presence of a third species.

Four-Species Mixed

The four-species mixed ecosystem is formed by adding an additional species to the three-species mixed system, such that both A and B have a cooperative partner. Here we find that B attains similar complexity to that of A in the three-species mixed system, and D attains similar complexity to that of C. In particular, A and B attain final complexities of 26.7 and 23.8 nodes, respectively, with highs of 29.9 and 26.2. C and D attain final complexities of 38.4 and 36.1 and highs of 39.8 and 38.0. We also see that species C and D sustain final complexity trajectories well above that observed in any other simulations, both with rates of more than 3 additional nodes per 1,000 generations.

Discussion

Our experiments reveal that co-evolution is able to drive complexity growth above that expected by genetic drift - indeed, all of our coevolutionary systems achieve higher final and maximal average complexity values than the control. However, the trajectories of complexity growth vary significantly depending on the nature of the coevolutionary interaction. Competitive co-evolution is able to drive extremely fast initial complexity growth, but then settles into stagnation. Cooperative co-evolution, somewhat surprisingly, is also able to drive complexity growth, but does so at a slower, steadier pace than competition.

Extension to three species systems reveals that a mix of competitive and cooperative relationships is a much more powerful driver of complexity growth than either competition or cooperation alone. Such a system seems to alleviate the phenomenon of mediocre stable states observed in two- and three-species purely competitive systems. Returning to the competing hypothesis of Gould and Dawkins, we find that genetic drift alone is able to produce complexity growth. However, this baseline growth is well below that observed in coevolutionary systems. The competitive arms races hypothesized by Dawkins are successful in creating complexity above that created by genetic drift, but fall short of other more successful systems.

Future Work

We’ve presented experimental results for a limited selection of the possible multi-species ecosystems. It may be worthwhile to perform a systematic analysis of all possible ecosystems of a given size, in the style of Wolfram’s analysis of elementary cellular automata. The dynamics of competitive systems have recently gained importance in the field of deep learning due to the success of generative adversarial networks (GANs) (Goodfellow et al., 2014). These systems often display the undesirable behavior observed in many purely competitive coevolutionary simulations, including those presented here. Much work has been done to coax these models towards more stable dynamics (Salimans et al., 2016; Arjovsky et al., 2017). We look towards applying the notion of mixed cooperative and competitive systems to the GAN architecture. A few works have successfully implemented systems which are approximately isomorphic to our three-species mixed model (Chen et al., 2016; Abadi and Andersen, 2016), although with the aim of extending the capabilities of the system, rather than improving learning trajectories.

Acknowledgements

The authors would like to acknowledge support for this work provided by the Alfred Schonwalter graduate fellowship in 2016.
References


