Reward-Based Social Learning

Wesley Tansey
Dept. of Computer Science,
The University of Texas at Austin
1 University Station C0500,
Austin, TX, USA
tansey@cs.utexas.edu

Eliana Feasley
Dept. of Computer Science,
The University of Texas at Austin
1 University Station C0500,
Austin, TX, USA
elie@cs.utexas.edu

Risto Miikkulainen
Dept. of Computer Science,
The University of Texas at Austin
1 University Station C0500,
Austin, TX, USA
risto@cs.utexas.edu

ABSTRACT

Social learning is an extension to evolutionary algorithms that enables agents to learn from observations of others in the population. Historically, social learning algorithms have employed a student-teacher model where the behavior of one or more high-fitness agents is used to train a subset of the remaining agents in the population. This paper presents a reward-based model of social learning in which agents are not labeled as teachers or students, instead allowing any individual receiving a sufficiently high reward to teach other agents to mimic its recent behavior. We validate our approach through a series of experiments in a robot foraging domain, including comparisons of reward-based social learning with simple neuroevolution and a variant of student-teacher social learning. The reward-based algorithm converges to near-optimal strategies faster than either benchmark approach, outperforming both in a complex foraging task by more than an order of magnitude. The resusts indicate that reward-based social learning is a promising new paradigm for social learning in intelligent agents.

General Terms

Social Learning, Evolutionary Algorithms, Artificial Life, Cultural Algorithms, On-line Evolution

1. INTRODUCTION

ed Evolutionary Algorithms (EAs) [6] evaluate agents either in isolation or in direct competition with a subset of the other members of the population. Social and cultural learning algorithms [14] extend EAs by enabling agents to leverage observations of other members of the population to improve their own performance during their lifetime. By learning from others without having to directly experience or acquire knowledge, social learning algorithms have been able to improve the learning rate of EAs in many challenging domains [5, 7, 18, 2, 4, 19].

Traditionally in social learning algorithms, each agent is either a student or a teacher.[**?**] All actions of the teacher agents are considered to be good examples from which to learn, as they are derived from a high-fitness strategy (i.e., the teacher's policy). However, an agent with high overall fitness may not always choose good actions. Conversely, agents with low overall fitness may actually perform well in some limited scenarios. Filtering potential observations based on their own merit therefore may be more appropriate and lead to both improved learning rate and stronger final strategies.

In this paper, we present a reward-based approach to social learning. Agents in our algorithm can learn from any other agent in their cultural group. The choice of learning examples is determined by a user-defined acceptability function that filters out examples leading to low rewards. When an acceptable action is observed, agents mimic that action in order to learn a policy similar to that of the observed agent. Our algorithm differs from other social learning algorithms in that the quality of a training example is measured by the reward received rather than the fitness of the agent generating the example.

We validate our algorithm in a well-known foraging domain in which agents must learn to navigate the world efficiently, gathering nutritious food while avoiding poisonous food. Our results indicate that reward-based social learning is able to quickly find near-optimal strategies more than an order of magnitude faster than a student-teacher variant and plain neuroevolution. Additionally, we show that our use of culture promotes behavioral diversity and prevents premature convergence to sub-optimal strategies that would otherwise emerge if agents observed all individuals in the population.

This paper makes the following contributions:

- A reward-based social learning algorithm, in which all individuals in a group observe and learn from, and teach each other;
- A behavioral diversity analysis of two reward-based social learning variants;
- A performance comparison between reward-based social learning and traditional student-teacher social learning.

The remainder of the paper is structured as follows. Background information and prior work is discussed in Section 2.

Section 3 details the how the reward-based social learning algorithm works. Section 4 describes both the experimental setup and the foraging domain in detail. Section 5 presents the results of the experiments. A discussion of the results and planned extensions to our work are described in Section 6. Finally, in Section 7 we present our conclusions.

2. BACKGROUND

In this section, we provide background information and motivation for social learning and introduce our evolutionary framework.

Social Learning

One explanation for the evolution of large brains in primates is the social intelligence hypothesis[**?**], which states that the need to handle complex social behaviors was the primary selection pressure driving the increase in brain size. The cultural intelligence hypothesis extends this concept solely to humans, stating that our brains evolved to handle the specific challenge of culture creation and social learning. These hypotheses are currently the most widely accepted explanations for the evolution of the human brain among evolutionary biologists and cognitive scientists [9], and have been supported by strong empirical evidence in recent years [8].

Cultural and social learning algorithms [14] model this biological mechanism in multi-agent systems by designating teacher agents that propagate knowledge and train other agents in the population. These techniques effectively enhance EAs with a hierarchical structure (i.e., students and teachers) that makes it possible to discover suitable actions to use as training examples and target individuals to train. Typical cultural algorithms therefore capture the ability of humans to learn from formal instruction by an expert, but they do not fully model all forms of learning from observation in primates. For instance, human brains contain mirror neurons [1] that activate when other humans are observed interacting with the world, in effect mirroring the observed human's action internally. Such processes enable humans to learn socially without concern for the status or expertise of the observed individual.

Related Work

Enhancing EAs with social learning is a flourishing area of research with a long and successful track record. We next highlight relevant prior work and explain how our approach differs from previous efforts.

Cultural algorithms [14] have been used frequently in Particle Swarm Optimization (PSO) [10]. These algorithms maintain a "belief space" representing different categories of knowledge that the population has learned. New individuals are trained using this belief space in a student-teacher paradigm. In contrast, our agents maintain no separate repository of formal knowledge, but rather they learn from observations of others during their lifetime.

Interestingly, our approach could be seen as a Memetic Algorithm (MA) [12]. As originally defined by Dawkins [3], a meme is a "unit of imitation in cultural transmission" including "tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches". Our approach particularly relates to the latter— ways of making pots or building arches are similar to strategies for gathering food in that they all are instances of transmitting specific strategies for performing skilled tasks. Thus, each reward-based learning example could be considered as a meme being shared within the agent's population or cultural group. Nevertheless, traditional MAs [13], including similar Lamarkian evolution methods for evolving recurrent neural networks [11], focus on using local improvement heuristics and typically rely on off-line transmission of memes via mating and genetic recombination. However, agents in our approach transmit memes on-line as they are generated and observed during the evaluation process. Thus, while we could classify our approach as an MA, we believe the distinct transmission and learning techniques employed by our algorithm better characterize it as a social learning method.

The ability of social learning to improve agents in a foraging domain has been explored by several researchers in recent years. Denaro et. al. [5] used a student-teacher model of cultural evolution without genetic inheritance and demonstrated that the population will continue to improve if Gaussian noise is added to the training examples. The NEW TIES system [7, 18] simulates a steady-state evolution of decision tree agents where at each step the teacher agents probabilistically transmit their decisions and students probabilistically incorporate this knowledge. Acerbi et. al. [2] use social learning to train embodied agents to mimic the behaviors of more experienced agents. Finally, de Oca et. al. [4] propose a methodology for incremental social learning in PSO to update Q-learning [19] value functions by randomly selecting two individuals from the population and combining their values for a given update. While all of these studies are closely related and motivated by similar biological processes as our approach, they fundamentally all rely on the concept of students and teachers, and perform either no filtering or a reward-agnostic filtering of state-action pairs to be used in updating the population.

Evolutionary Framework

NeuroEvolution of Augmenting Topologies (NEAT)[17] is an evolutionary algorithm that generates recurrent neural networks. Through a process of adding and removing nodes and changing weights, NEAT evolves genomes that unfold into networks. In every generation, those networks with the highest fitness reproduce, and with the lowest fitness are unlikely to do so. NEAT maintains genetic diversity through speciation and encourages innovation through explicit fitness sharing.

In our domain, NEAT is used to generate a population of individual neural networks that control agents in the world. The input to each network is the agent's sensors, and the outputs control the agent's velocity and orientation. The fitness of each network is determined by the success of the agent it controls - over the course of a generation, networks that control agents who eat a good deal of rewarding food and very little poison will have high fitnesses and those that control agents with less wise dietary habits will have low fitness.

In standard NEAT, the networks that are created do not change within one generation, but in reward-based social learning, we do backpropagation [15] on the networks that

NEAT creates. Since NEAT networks are recurrent, we use backpropagation through time [20] as our training algorithm. The final fitness of each phenome, then, reflects the performance of the individual that used that phenome and elaborated on it over the course of a generation. This elaboration drives evolution in two alternate ways. In Darwinian evolution, the changes that were made to the phenome only affect selection and are not saved; in Lamarkian, the genome itself is modified.

3. REWARD-BASED SOCIAL LEARNING

Algorithm 1 presents the on-line, generational learning algorithm for reward-based social learning. The algorithm is designed for domains where an agent receives multiple rewards throughout its lifetime. Agents maintain a sliding window of their most recent states, actions, and rewards that serve as training examples for observing agents. When an agent receives a reward, other agents observe the event and determine if it qualifies as an acceptable training example. Our key insight is that the agents should focus on the event itself to determine if the observed actions should be emulated. The overall fitness of the acting agent is not considered, as it is possible that a low-fitness strategy may perform high-quality actions in certain circumstances.

To promote diversity and prevent the population from prematurely converging to a local optimum, all agents are divided into observation groups called *cliques*. At each timestep, each agent observes its state, responds with a desired action, and receives a resulting reward. The memory of an individual agent is thus a tuple containing its most recent state, action, and reward. Each agent's memory is evaluated by all other members of its clique using a user-defined acceptability function. The role of this function is to filter out low-quality examples and focus the agents on learning only from actions that will lead to strong strategies. When an agent's recent memory is judged as acceptable, it is then used as a teaching example for all other agents in its clique.

The next section presents the experimental domain used to evaluate the efficacy of reward-based social learning.

Algorithm 1 Reward-Based Social Learning

```
Input: generations, maxTimesteps
population ←InitializePopulation()
g ← 0
t ← 0
loop
  while g < generations do
    cliques ← FormCliques(population)
    while t < maxTimesteps do
      for each clique in cliques do
        for each agent in clique do
          s ←ReceiveAgentInputs()
          a ←GetResponse(agent, s)
          r ←TakeAction(action)
          agent.memory ←UpdateMemory({s, a, r})
        end for
        for each agent in clique do
          if Acceptable(agent.memory) then
            for all observer in clique do
              Train(observer, agent.memory)
            end for
          end if
        end for
      end for
      t ← t + 1
    end while
    population ←SelectAndReproduce(population)
    g ← g + 1
  end while
end loop
```

4. EXPERIMENTAL SETUP

This section we describes the evaluation domain, including the inputs and outputs that agents receive, and the common parameters across all experiments.

The Foraging Domain

Our domain is a foraging world in which agents move freely on a continuous toroidal surface. The world is populated with various plants, some of which are nutritious and bear positive reward, while others are poisonous and bear negative reward. These plants are randomly distributed over the surface of the world. The foraging domain is non-competitive and non-cooperative; each agent acts independently of all other agents, with the exception of the teaching signals that pass between them. At the start of each generation, all individuals begin at the center of the world, oriented in the same direction, and confronted with the same plant layout and configuration. Every agent then has a fixed number time steps to move about the surface of the world eating plants— which happens automatically when an agent draws sufficiently close to one— before the evaluation is over. Figure 1 shows an example of a foraging world with two types of plants and eight agents.

Sensors and Outputs

Agents "see" plants within a 180° horizon via a collection discretized sensors. Each agent has eight sensors for each type of plant, with each sensor covering a different 12.5° sector of the 180° ahead of the agent. Agents cannot see other individuals, or plants they have already eaten— all they can see is edible food. The strength of the signal generated by each plant is proportional to its proximity to the agent. Agents also have a sensor by which they can detect their current velocity. As agents can only turn up to 30°, knowledge of velocity enables agents to accurately predict the effect of their actions. Each agent is controlled by an artificial neural network that maps from the agent's sensor readings to the desired change in orientation and velocity. Figure 2 shows the architecture of a foraging agent.

Common Parameters

In our experiments, we used two separate configurations for the robot foraging world. In the first three experiments, we used a "simple" world where the toroidal surface is 2000 by 2000 units, with a single plant type of value 100 and 50 randomly distributed instances of the plants. In this world, the agents have a straight-forward task of learning to navigate efficiently and gather as many plants as possible.

For our fourth set of experiments, we evaluated using both

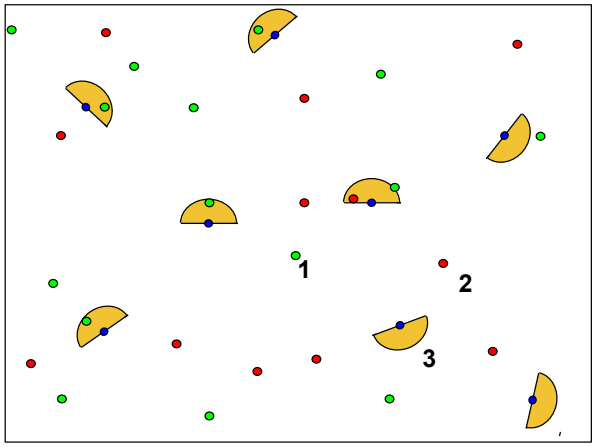

Figure 1: Our domain: a foraging world in which agents gain fitness by consuming plants which they approach. 1) a piece of nutritious food which increases fitness, 2) is poison which decreases fitness, and 3) is one of our agents.

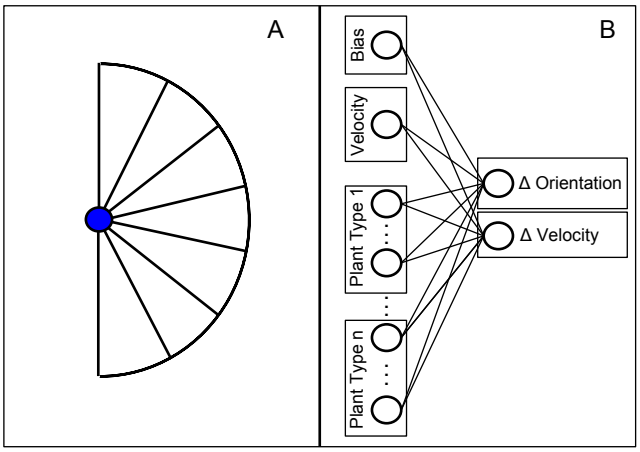

Figure 2: The architecture of our foraging agents. A) Each agent has a 180° field of vision, discretized into eight sensors for each plant type. B) Each agent is controlled by a neural network taking the agent's current velocity and sensor activations as inputs and outputting the desired change in orientation and velocity.

the simple world and a second, more complex world. The "complex" world has a surface of 500 by 500 units, with five different plant types of value -100, -50, 0, 50, and 100. For each plant type, we create 20 randomly distributed plant instances. This world presents the agents with a more difficult task as they must both efficiently gather nutrious food while simultaneously avoiding the poisonous food.

In all our experiments, we create 100 different agents in each generation. All networks are initialized with fully-connected weights with no hidden neurons. We use a learning rate of 0.1 when performing backpropagation. Agents automatically eat any plant with which they are within five units. Each evaluation lasts 1000 timesteps and our results for each experiment are the average of 30 independent runs. The filtering function for all experiments is to learn from any action yielding a positive reward.

We next describe our experiments and their outcomes in detail.

5. RESULTS

In this section we present our experimental results. We conducted four main experiments, 1) determining the baseline performance of reward-based social learning and neuroevolution, each in isolation, 2) measuring the performance of reward-based learning in Darwinian and Lamarkian evolutionary paradigms, 3) evaluating whether cultural learning improves performance compared to having a mono-culture population, and 4) comparing reward-based learning to a student-teacher learning algorithm in both simple and complex foraging tasks. The first three experiments examine the efficacy of the various components of our algorithm while the fourth experiment serves as a validation for our final algorithm against a variant of a related approach in the literature. The results of our experiments show that our algorithm is able to evolve high-quality strategies with substantially fewer agent evaluations than previous algorithms.

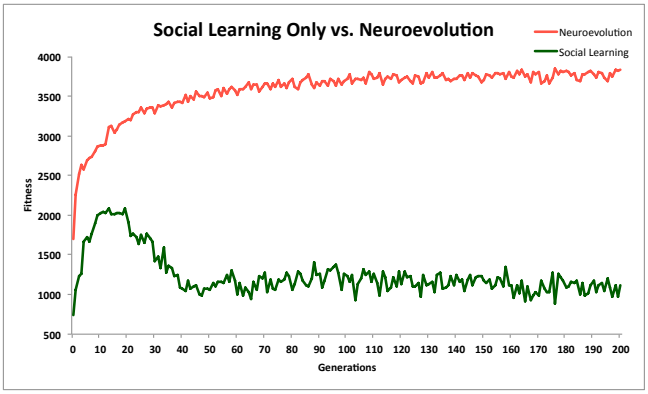

Figure 3: Pure reward-based learning collapses after several generations, while neuroevolution alone converges to a stable but suboptimal solution.

Reward-Based Learning vs. Neuroevolution

We first wish to establish a baseline for understanding the ability of our two component methods, social learning and neuroevolution, to evolve high-quality strategies in isolation. To evaluate social learning in isolation, we created a population of 100 randomly initialized neural networks and evaluated them without selection or reproduction for 200K time steps. After every 1000 time steps, the population champion was recorded and this was treated as a "generation" in order to compare it to baseline neuroevolution.

Figure 3 shows the results of the experiment. While reward-based social learning alone is able to improve initially, after several epochs a regression-to-the-mean effect is observed in which the entire population converges to a mediocre average score. A similar effect was observed in previous social learning experiments [5], suggesting that some form of mutation is needed to prevent population collapse. In contrast,

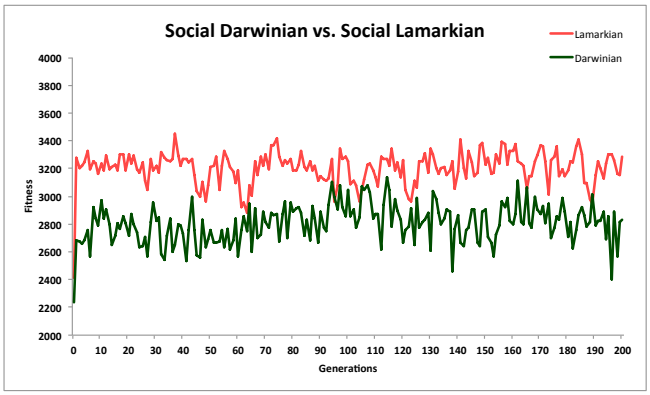

Figure 4: Lamarkian learning is more effective than Darwinian and converges more rapidly.

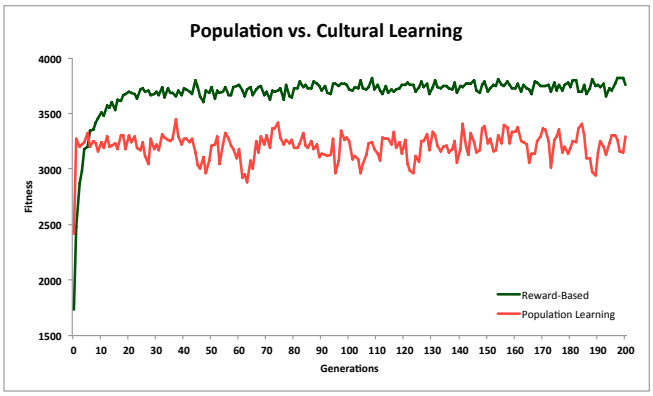

Figure 5: The results of agents learning from observations of the entire population compared to only agents in the same clique.

neuroevolution converges to solutions with fitnesses two to three times higher than that of reward-based social learning alone after approximately 50 generations.

Having validated that social learning alone is not sufficient, we next compare the performance of social learning when used as an enhancement to evolution.

Darwinian vs. Lamarkian Evolution

Genetic inheritance paradigms in evolution fall into one of two main categories: Darwinian and Lamarkian. In Darwinian evolution, individual genomes are fixed and any knowledge or abilities gained during their lifetimes are not passed on to their offspring at birth. By contrast, in Lamarkian evolution an individual's genome changes as it learns throughout its life, and these changes are passed on to each of its offspring. In the context of our experiments, this corresponds to whether the changes in each individual's neural network weights, as a result of social learning, are propagated to their genome at the end of the generation.

Figure 4 shows the results of applying our reward-based social learning algorithm to the foraging domain in both the Lamarkian and Darwinian paradigms. The performance of both algorithms quickly converges, with Lamarkian reaching a higher solution than Darwinian evolution. In the context of *on-line* evolutionary learning algorithms, previous work [21] showed that Darwinian evolution is likely to be preferable to Lamarkian evolution in dynamic environments where adaptation is essential and the Baldwin effect [16] may be advantageous. However, as adaptation is not necessary for our agents (i.e., the rewards of each plant type are the same in every generation), in this experiment Lamarkian evolution outperforms Darwinian evolution.

Nevertheless, in both cases performance converges to a lower score than that of simple neuroevolution. We next present the results of introducing culture into reward-based social learning to overcome such convergence.

Population vs. Cultural Learning

On one hand, Lamarkian social learning is able to find good results quickly, while on the other hand, it is likely to provide redundant information that may result in getting stuck in local optima.

In order to address this problem, we introduce a subcultural version of reward-based social learning designed to promote and protect diversity. At the start of each generation, the population is divided into 10 cliques of 10 agents each, with each agent's clique decided at random. During the evaluation, agents only teach and learn from other agents in their own clique. In addition to increasing behavioral diversity relative to learning from the entire population, this variant also has the appealing practical advantage that it decreases the worst-case number of iterations of backprop. For instance, in a population of 100 individuals with ten subcultures, an accepted training example is propagated to only one eleventh as many agents as it would be in population-wide learning.

Figure 5 shows the results comparing population-based and cultural learning. The cultural social learning not only reaches a higher peak than the population-based method, but also arrives at this level of fitness more rapidly than the baseline neuroevolution approach. Preventing agents that lead the population towards a local optimum from impacting the remainder of the population provides safety and protection; when every mutated organism has the opportunity to train every other, as is the case in population-wide learning, the entire population may be negatively impacted by such individuals. Because cultural evolution is more efficient and provides better results we incorporate it in our final algorithm.

To better understand why the cultural version of our algorithm was able to prevent premature convergence, we analyzed the behavioral diversity of the population after every generation. After each evaluation, we generated a new instance of the simple world and created a set of input states corresponding to random locations, orientations, and velocities. For every agent, we then observed its output response (i.e., the desired change in velocity and orientation) for each of the input states. Figures 7 and 6 show the average variance of the population response to each of the 100 input states after every generation, effectively measuring the behavioral diversity of the population. Velocity response con-

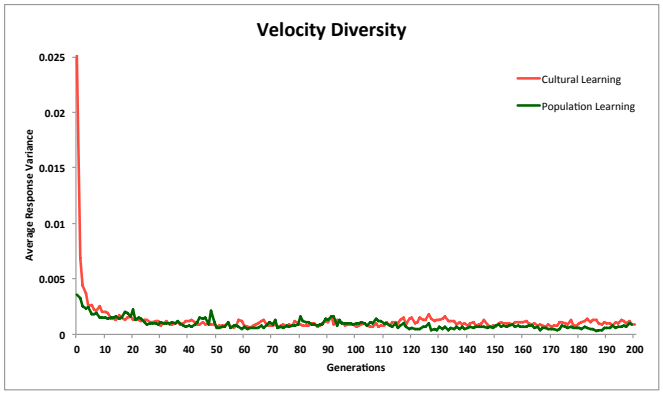

Figure 6: Velocity converges more quickly in the cultural model than in the population model.

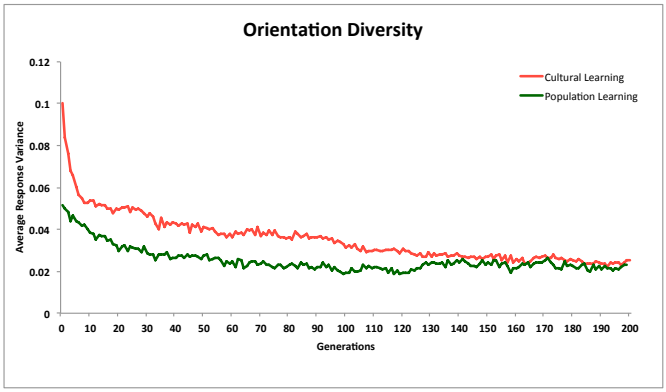

Figure 7: In the cultural learning model, diversity in terms of orientation is maintained for longer than in population learning before behavior converges.

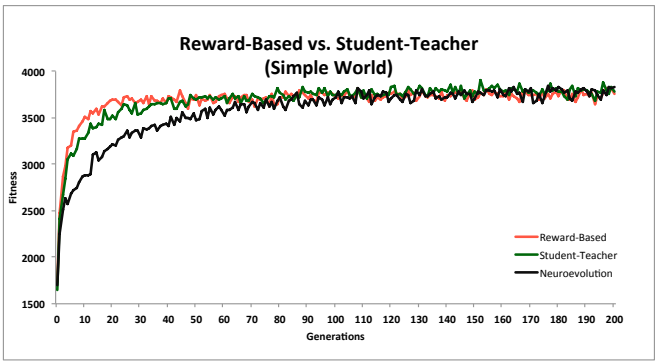

Figure 8: The results of our algorithm in a simple world evaluated alongside a more traditional student-teacher model and baseline pure neuroevolution.

verges for both models almost immediately, while orientation response remains more diverse in subcultural than in population-wide learning.

Although the orientation and velocity results may seem contradictory, they actually reflect a desirable property of subcultural evolution. In the case of velocity, both the population and cultural algorithms quickly converge to low diversity. Since there is no benefit to slowing down in our current domain, one would expect a strong strategy to always output the maximum value for velocity. Thus, preserving diversity in this area would be detrimental to the performance of the population. Conversely, the diversity of orientation should be preserved, as there is a tradeoff between turning toward an immediate but small reward, such as a single plant within one step, and turning toward a delayed reward, such as a cluster of plants a few steps away. This complexity suggests that the fitness landscape for orientation is likely filled with numerous local optima and diversity preservation should be beneficial to evolution.

Thus the results of the diversity analysis suggest that the subcultural algorithm preserves diversity only when it is useful (e.g., orientation response), while not preventing convergence when diversity is unnecessary (e.g., velocity response).

The next experiment compares our cultural learning algorithm to a benchmark student-teacher model.

Reward-Based vs. Student-Teacher

In our final set of experiments, we compare our cultural reward-based algorithm to an on-line student-teacher learning algorithm inspired by the NEW TIES system [7]. The system utilizes a steady-state evolution in which at each agent probabilistically teaches the lowest-fitness member of the population within some radius at every timestep, effectively forming geographical subcultures. Since the system relies on a steady-state evolution, weaker agents eventually die out and do not have as much of a chance to propagate poor training examples as long-lasting agents. To adapt this approach to our generational EA, and to maximize the fairness of our comparison, we create a population of 10 cliques with 10 agents each. At each time step, the current

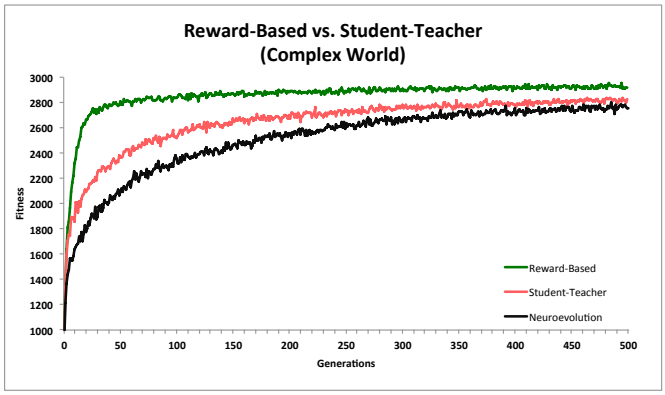

Figure 9: The results of our algorithm in a complex world evaluated alongside a more traditional student-teacher model and baseline pure neuroevolution.

champion from each clique teaches its clique's lowest-fitness member.

It is worth noting that many social learning algorithms focus on off-line training, where teachers are simply the previous generation's champions. In these models, students are trained before the actual evaluation to mimic the champion(s), often with Gaussian noise added to the teacher's outputs to promote diversity in the population. [5] However, at evaluation time these agents are static and do not learn from their own experiences or that of others. Thus, we chose to implement NEW TIES in a generational variant as it is the most closely related approach in the literature.

Figures 8 and **??** show the results of our cultural reward-based learning algorithm compared to our student-teacher variant and pure neuroevolution. Reward-based learning converges to a near-optimal solution faster than the student-teacher variant in both the simple and the complex world. While in the simple world (Figure 8) this speed-up is slight, in the complex world (Figure **??**) our reward-based approach is more than an order of magnitude faster, reaching a higher fitness by generation 50 than either student-teacher or baseline neuroevolution achieve by generation 500.

In the next section, we present a discussion of our experimental results and indicate possible future directions for reward-based social learning.

6. DISCUSSION AND FUTURE WORK

The results of our experiments in reward-based social learning suggest that learning from strong actions, rather than high-fitness agents, is a powerful approach to social learning. The success of our algorithm in a given domain is predicated on two main user-defined parameters: the grouping of agents into cliques and the criteria of the acceptability function.

For our experiments, we chose a simple division of 10 groups of equal size for our clique creation strategy. Preliminary experiments showed that our algorithm was robust to moderate variations in the number of equal-sized cliques, with 5 groups and 20 groups both demonstrating strong speedup compared to our benchmarks. However, it's possible that equal-sized groups are not optimal or that a dynamic clique creation algorithm that maximizes behavioral diversity would perform better; such issues will be addressed in future work.

Creating an acceptability function is a closely related task to that of creating a fitness function. Consequently, the performance of our algorithm depends on the ability of the user to craft a useful acceptability function. This task is not always easy and may require paying attention to the subtle details of one's domain. We intentionally kept the acceptability function simple in our domains so as to demonstrate the strength of our general approach, however it is possible that employing a more sophisticated heuristic could speed up learning even more.

The robot foraging domain is well-suited for exploring extensions and improvements to our social learning algorithm. Whereas the current version only utilizes a single example at every time step, teaching more complex behaviors using reward-based approaches may require larger memory sizes. One interesting future work direction may be to introduce obstacles or walls that block the agent from immediately receiving the reward, necessitating that agents learn compound trajectories to navigate to the food.

Finally, we plan to expand our benchmark suite to include more student-teacher variants in the future. Although we believe our experiment sufficiently demonstrated the advantages of reward-based social learning, student-teacher methods can be implemented in many ways, including off-line learning [2], probabilistic teaching [7], and noisy examples [5], to name only a few. A more complete set of benchmarks will help us better understand the relative performance of each approach.

7. CONCLUSIONS

We have presented a reward-based approach to social learning in evolutionary algorithms that enables agents to learn from high-quality actions. As opposed to traditional social learning algorithms that follow a student-teacher model, our approach instead teaches agents based on acceptable actions taken by any agent in its clique. By constraining teaching samples to those from the same cultural group, our algorithm promotes diversity in the overall population and prevents premature convergence. Experiments in a complex robot foraging domain demonstrated that this approach is highly effective at quickly learning a near-optimal policy with Lamarkian evolution. Our results suggest that reward-based learning is a strong technique and represents a promising new paradigm for social learning algorithms.

8. REFERENCES

[1] Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493 – 501, 1998.

[2] A. Acerbi and S. Nolfi. Social learning and cultural evolution in embodied and situated agents. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pages 333–340. IEEE, 2007.

[3] R. Dawkins. *The selfish gene*. Oxford University Press, 1976.

[4] M. de Oca, T. Stutzle, K. Van den Enden, and M. Dorigo. Incremental social learning in particle swarms. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(2):368–384, 2011.

[5] D. Denaro and D. Parisi. Cultural evolution in a population of neural networks. *M. marinaro and r. tagliaferri (eds), neural nets wirn-96. new york: Springer*, pages 100–111, 1996.

[6] L. Fogel, A. Owens, and M. Walsh. Artificial intelligence through simulated evolution. 1966.

[7] E. Haasdijk, P. Vogt, and A. Eiben. Social learning in population-based adaptive systems. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1386–1392. IEEE, 2008.

[8] E. Herrmann, J. Call, M. Hernández-Lloreda, B. Hare, and M. Tomasello. Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *science*, 317(5843):1360, 2007.

[9] K. Holekamp. Questioning the social intelligence hypothesis. *Trends in cognitive sciences*, 11(2):65–69, 2007.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[11] K. Ku, M. Mak, and W. Siu. A study of the lamarckian evolution of recurrent neural networks. *Evolutionary Computation, IEEE Transactions on*, 4(1):31–42, 2000.

[12] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826:1989, 1989.

[13] G. Onwubolu and B. Babu. *New optimization techniques in engineering*, volume 141. Springer Verlag, 2004.

[14] R. Reynolds. An introduction to cultural algorithms. In *Proceedings of the Third Annual Conference on Evolutionary Programming*, pages 131–139. World Scientific, 1994.

[15] D. Rumelhart, G. Hintont, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[16] G. Simpson. The baldwin effect. *Evolution*, 7(2):110–117, 1953.

[17] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[18] P. Vogt and E. Haasdijk. Modeling social learning of language and skills. *Artificial Life*, 16(4):289–309, 2010.

[19] C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[20] P. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[21] S. Whiteson and P. Stone. Evolutionary function approximation for reinforcement learning. *The Journal of Machine Learning Research*, 7:877–917, 2006.