Non-hierarchical Social Learning via Reward-Based Update Filtering

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

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

ABSTRACT

Social learning is an extension to Evolutionary Algorithms that enables individuals to learn from observations of others in the population. Traditionally, social learning algorithms have employed a student-teacher model where the behavior of one group of individuals is used to train the remaining individuals in the population. We present a non-hierarchical model of social learning in which we do not label each agent, instead allowing any individual which experiences positive reward to teach the rest of the agents on its recent behavior. We validate our approach in a foraging domain, comparing social learning in both Darwinian and Lamarkian paradigms to a standard Darwinian evolution with no learning. We show that our non-hierarchical form facilitates rapid discovery of near-optimal solutions. While Lamarkian evolution eventually produces a regression-to-the-mean effect, we bootstrap several generations of Lamarkian evolution with regular GAs to produce a highly efficient solution to the foraging problem.

General Terms

Social Learning, Evolutionary Algorithms, Artificial Life

1. INTRODUCTION

One explanation for the evolution of large brains in primates is the social intelligence hypothesis, which states that the selection pressure driving the increase in brain size was the need to handle complex social behavior. The cultural intelligence hypothesis extends this concept specifically to humans, stating that our brains evolved to handle the specific challenge of culture creation and social learning. As well as being an intuitive justification, these hypotheses are currently the most widely accepted explanations for the evolution of the human brain among evolutionary biologists and cognitive scientists [**?**], and has been supported by strong empirical evidence in recent years [**?**].

Cultural and social learning algorithms [**?**] 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 Evolutionary Algorithms (EAs) with a hierarchical structure (i.e., students and teachers) that facilitates the automatic discovery of suitable actions to use as training examples and target individuals to train. Thus, while cultural algorithms capture the ability of humans to learn from formal instruction, they do not fully model all forms of learning from observation in primates.

We present a non-hierarchical approach to social learning, inspired by mirror neurons [**?**], where agents learn by observing the actions of other agents. Primate brains contain mirror neurons that activate when seeing other primates carry out an action, in effect mirroring the observed primate's action internally. Analogously, agents in our algorithm observe the population and, when a positive reward is received, mimic that action in order to learn a policy similar to that of the observed agent. This algorithm separates itself from other social learning algorithms in that the quality of a training example is measured by the reward received rather than the role of the observed agent.

We validate our algorithm in a well-known foraging domain in which agents must discriminate between poisonous and nutritious food. Through our experiments in this domain, we compare our non-hierarchical social learning approach in both a Darwinian and Lamarkian evolutionary paradigm. By using social learning, the individuals in our population were able to achieve near-optimal performance in many fewer generations than were agents that did not learn during a lifetime and only evolved. Our results further indicate that a social Lamarkian bootstrapping phase not only drastically speeds up learning but also increases the maximum fitness of the champion individuals.

This paper makes the following novel contributions:

- A non-hierarchical approach to social learning, in which individuals are not classified to be teachers and students.
- A hybrid social learning algorithm that uses social learning as a bootstrapping phase for neuroevolution.
- An analysis of the differences in performance between Darwinian and Lamarkian evolution in the place of social learning.

The remainder of the paper is structured as follows. In Section 2 we detail the workings of our non-hierarchical social learning algorithm. In Section 3 we describe our experimental setup and the foraging domain. Section 4 presents the results of our experiments. Related work is discussed in Section 5. Planned extensions to our work is described in Section 6. Finally, in Section 7 we present our conclusions.

2. NON-HIERARCHICAL SOCIAL LEARNING

In this section, we elaborate on our approach and its justification and applications. First, we discuss some of the advantages of non-hierarchical social learning and the domains in which it provides promising functionality. Next, we describe the algorithm at the core of our model, based on positive rewards. Finally, we go into some detail about the particulars of our implementation.

Social learning is valuable in expensive domains; when computing time is limited or individuals have limited experiential training data, leveraging the experiences of multiple individuals is a valuable way to use the information that is available. As video games and real-life applications of intelligent agents become more pervasive, these expensive-to-simulate, easy-to-record domains are becoming more and more common. Every agent can benefit from the experiences of every other without the expense of rerunning the training environment.

Another important area for social learning is dynamic domains. In multiagent systems, changing conditions can negatively impact all agents if they cannot learn from one another's experience - but if they can, one agent's experience of a changing condition can alert other agents so they can adjust their behavior. If a social learning system is online, sharing updates and information about reward at every timestep, adaptation can occur rapidly.

Our approach, detailed in 1 is an on-line learning algorithm that operates continuously as an agent moves in the world. At every timestep, each agent perceives the state of the world around it, and activates an internal neural network according to that state. The output of this network represents the agent's motor commands - what actions it should take. Both the input and the output of the network are saved and stored in memory. Upon moving, an agent encountering a positive reward will retreive its recent inputs and associated outputs from memory and train other individuals on these input-output pairs using backpropagation.

As a result of using only positive rewards to identify those actions on which we want to train agents, it takes only the notion of a reward to allow our social learning agents to begin to teach one another on line. Previous approaches [?] instead chose strong individuals as teachers to train weaker individuals in appropriate behavior. By allowing any individual to train any other, we leverage a diversity of different behaviors in solving problems in a domain.

Evolutionary Framework

In this section, we discuss the framework in which we based our social learning, NeuroEvolution of Augmented Topologies (NEAT)[?], and our modifications and extensions to this framework.

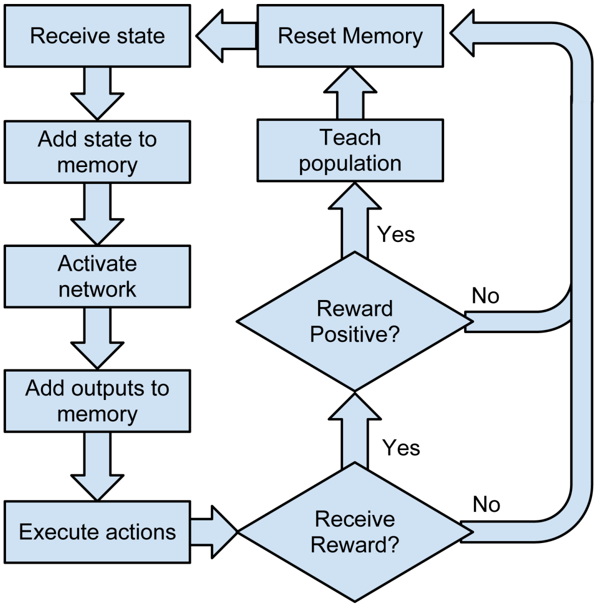

Figure 1: Individuals remember their most recent inputs and the associated actions taken, and when rewarded will train other individuals on their recent actions.

NEAT is an evolutionary algorithm that generates recurrent neural networks. Through a process of adding and removing nodes and changing weights, NEAT evolves phenomes that unfold into networks. In every generation, those networks with the highest fitness continue, and those that have low fitness are less likely to do so. NEAT maintains diversity in a number of ways, notably through maintaining a variety of different species, neural network phenomes that are related.

In our domain, NEAT is used to generate a population of individual neural nets that control agents in the world. The input to each network is the agent's sensors, and the output is two controls, of which one accelerates and decelerates the agent and the other turns it. The fitness of each network is determined by the success of the agent it controls - over the course of 1000 timesteps, networks controlling agents who eat a good deal of rewarding fruit and very little poison will have high fitnesses and those which 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 non-hierarchical social learning, we do backprop on the networks that NEAT creates. (Because these networks are recurrent, we use backpropagation through time to do our social learning [?].) 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. In Darwinian evolution, the changes that were made to the phenome over the course of a generation are not saved; in Lamarkian, the phenome itself is modified.

3. EXPERIMENTAL SETUP

In this section we first describe the domain in which we test our model, and discuss the workings of the agents themselves. Next, we describe the experiments, before discussing results in the next section.

The Foraging Domain

Our domain is a foraging world in which agents move freely on a continuous toroidal surface. We populate our world with various plants, some of which bear positive reward that increases an agents' fitness, and others of which bear negaive reward which reduces it. These plants are randomly distributed over the surface of the world. This foraging domain is non-competitive and non-cooperative - each agent acts independently of all other agents, with the exception of the training signals which pass between them. Each individual begins each generation in the center of the world, oriented in the same direction, and confronted with the same plants. Every agent then has several time steps to move about the surface of the world eating plants - which happens automatically when an individual draws close - before the generation is over and a new population is evolved.

Sensors and Outputs

Agents "see" plants within a certian horizon via a collection of sensors - they have several sensors for each type of plant, each of which detects plants in a different sector of the 180 degrees 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 coming into a sensor is proportional to both the proximity of the plant it is detecting and the number of plants visible. Agents also have a sensor by which they can detect their current velocity. These sensors constitute the inputs to the neural network, and the individual's state.

Each agent has two controls that are the outputs of the neural network - an accelerator/decellerator which changes the agent velocity, and another control that turns the agent to the left or right. Each of these output nodes is mapped onto a range for the values of these controls. The agent never knows its absolute orientation in the world or its absolute position - it must make decisions based only on the input from its sensors.

Common Parameters

For all of our experiments, the world is 500 by 500 units, with 20 randomly distributed plants of each value -100, -50, 0, 50, and 100. We create 100 different agents in each generation, and speciate them into 10 species. Each agent had 8 different sensors for each type of plant, and automatically ate any plant it came within 5 units of. Each generation lasted 1000 timesteps, and each experiment was averaged over 30 runs.

4. RESULTS

We next present the results of our three experiments that validate our non-hierarchical model. We begin by measuring the performance of social learning in both Darwinian and Lamarkian evolutionary paradigms. Following this, we determine whether social learning is effective when updates are performed only among the same species, with the goal of reducing the overall runtime cost of adding social learning.

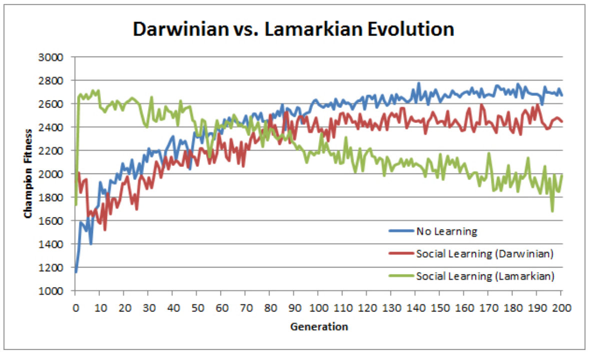

Figure 2: A comparison of the results for our non-hierarchical social learning algorithm in Darwinian and Lamarkian evolutionary paradigms.

Finally, leverage the insights from the first two experiments and use Lamarkian social learning as a bootstrapping phase for Darwinian neuroevolution with no learning.

Darwinian vs. Lamarkian

Genetic inheritance paradigms in evolution falls into one of two main categories: Darwinian and Lamarkian. In Darwinian evolution, individual genomes are fixed and any knowledge gained 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 at birth. In the context of our experiments, this corresponds to whether the changes in each individual's neural network weights are propagated to their genome at the end of the generation.

Figure 2 shows the results of applying our non-hierarchical social learning algorithm to the foraging domain for both the Lamarkian and Darwinian paradigms. These results indicate that Lamarkian evolution is able to quickly reach a near-optimal score but then proceeds to degrade slowly over time. In the context of *on-line* evolutionary learning algorithms, it has been shown [?] that Darwinian evolution is preferable to Lamarkian evolution in dynamic environments where adaptation is essential and the Baldwin effect [?] may be advantageous. As adaptation is not necessary for our agents (i.e., the rewards of each plant type are the same in every generation), it is not exceedingly surprising that Lamarkian evolution outperforms Darwinian evolution initially.

However, the degradation of the Lamarkian fitness after generation 10 is surprising. We believe this is likely due to a "regression to the mean" effect where once the population reaches a sufficiently high fitness, the learning is derived more from the average individuals and less from the best individual. Thus, rather than the best individual pulling the other individuals' fitness up, the average individuals actually begin to pull the population as a whole down. A similar effect has been observed before [?].

Population vs. Species Learning

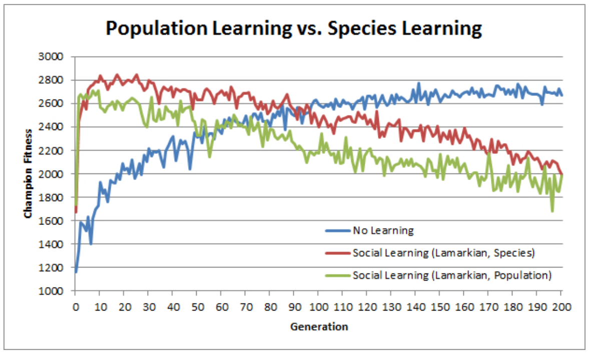

Figure 3: The results of agents learning from observations of the entire population compared to only agents in the same species.

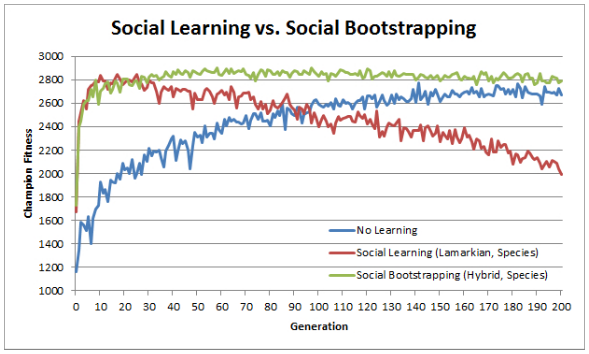

Figure 4: The results of our hybrid algorithm that uses social Lamarkian evolution to bootstrap the population for five generations then switches to traditional non-social Darwinian evolution.

While Lamarkian social learning is clearly able to find results quickly, it suffers from two main issues. As discussed above, the population begins to regress towards the mean after reaching the initial peak fitness. Also, while in many environments the simulation time required for running thousands of backprop on each individual may be irrelevant, to maximize practical efficiency it is important that our algorithm minimizes its overall impact on total runtime. To address these issues, we next consider a cultural variant of our social learning approach in which individuals only learn from observations of other individuals in the species. In practice, this significantly speeds up the application as it performs an order of magnitude less work for a population of 100 agents divided into 10 species.

Figure 3 shows the results comparing population-based and species-based social learning. Interestingly, the species-based social learning not only reaches a higher peak than the population-based method, but is also able to sustain its level of fitness for longer. Unfortunately, both approaches still suffer from the degradation of fitness characteristic of Lamarkian social evolution.

Bootstrapping

The ability to find a near-optimal fitness combined with the subsequent degradation of individuals in later generations suggests that social Lamarkian evolution may be best applied only in the initial generations. Figure 4 presents the results of a hybrid approach that uses social Lamarkian evolution for the first five generations to bootstrap the population, then transitions to the tradition non-social Darwinian evolution. The hybrid version is able to achieve a slightly higher (though not statistically significant) fitness than either comparison method and does not suffer from the degradation present in the pure social Lamarkian setup.

In the next section we present a brief discussion of related work on social learning in EAs.

-

5. RELATED WORK

Enhancing EAs with social and cultural 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 [?] have been used frequently in Particle Swarm Optimization (PSO) [?]. Cultural 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.

The ability of social learning to improve agents in a foraging domain is a popular setting that has explored by several researchers in recent years. Denaro et. al. [?] 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 [?, ?] 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. [?] use social learning to train embodied agents to mimic the behaviors of more experienced agents. Finally, de Oca et. al. [?] propose a methodology for incremental social learning in PSO to update Q-learning [?] value functions by randomly selecting two individuals from the population and combining their values for a given update. While all of these works 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.

6. FUTURE WORK

In this section we discuss potential future work that could be done to exploit non-hierarchical social learning and improve it as both a model of artificial life and a machine learning algorithm. Future work may involve investigating the relationship between this and other forms of social learning and Q-learning. Additionally, one strength of non-hierarchical social learning is its ability to transmit information about novel situations to all agents without those agents having to experience those situations themselves. As such, investigating the impact of non-hierarchical social learning in dynamic domains with changing rewards is a promising and practical avenue for new research. The current non-hierarchical social learning model teaches agents about the previous timestep with one iteration of backprop whenever there is a reward. Improving the model to account for the magnitude of the reward, and to store and train on information about previous timesteps may lead to new insights. Finally, the foraging domain in its current incarnation is too easy for our algorithm - Lamarkian evolution solves it within a few generations. Non-hierarchical social learning should be applied to more difficult problems.

7. CONCLUSIONS

We have presented a non-hierarchical approach to apply social learning in evolutionary algorithms. Traditionally, social learning in evolutionary algorithms has followed a student-teacher model that assigns roles to each agent. Our approach removes this hierarchy and instead updates individuals based on actions taken by *any* agent that lead to a positive reward. Experiments in a complex robot foraging domain demonstrate that this approach is highly effective at quickly learning a near-optimal policy with Lamarkian evolution. Further results from our hybrid algorithm suggest that bootstrapping a traditional Darwinian EA with a brief period of non-hierarchical Lamarkian social learning can substantially improve performance of the baseline EA and reaches higher fitness than either approach in isolation.

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] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

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

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

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

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

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

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

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