Curriculum Learning for Tightly Coupled Multiagent Systems

Extended Abstract

Golden Rockefeller
Oregon State University
rockefeg@oregonstate.edu

Patrick Mannion
Galway-Mayo Institute of Technology
patrick.mannion@gmit.ie

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

ABSTRACT

In this paper, we leverage curriculum learning (CL) to improve the performance of multiagent systems (MAS) that are trained with the cooperative coevolution of artificial neural networks. We design curricula to progressively change two dimensions: scale (i.e. domain size) and coupling (i.e. the number of agents required to complete a subtask). We demonstrate that CL can successfully mitigate the challenge of learning on a sparse reward signal resulting from a high degree of coupling in complex MAS. We also show that, in most cases, the combination of difference reward shaping with CL can improve performance by up to 56%. We evaluate our CL methods on the tightly coupled multi-rover domain. CL increased converged system performance on all tasks presented. Furthermore, agents were only able to learn when trained with CL for most tasks.

KEYWORDS

Curriculum learning; multiagent coordination; difference rewards

ACM Reference Format:
Golden Rockefeller, Patrick Mannion, and Kagan Tumer. 2019. Curriculum Learning for Tightly Coupled Multiagent Systems. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Reinforcement learning in tightly coupled tasks in complex multiagent systems (MAS) is difficult. In such tasks, a team of agents must together be in the right place and select the right action at the right time to achieve their goal and receive a reward. In addition, due to the "curse of dimensionality" [2], the moments in which pre-trained agents are able to successfully coordinate their actions through random chance are rare. As a result, learning in MAS with tight coupling presents a causality dilemma; one agent can not learn unless other agents are doing the right thing and other agents will not do the right thing without learning.

The solution we present in this paper is the application of curriculum learning (CL). CL extends the idea of transfer learning [9] by training agents on progressively more complex versions of a specific task, i.e. a curriculum. By using prior knowledge to initialize agents' policies before the learning process, we aim to circumvent the causality dilemma by increasing the likelihood that some agents will do the right thing from the beginning of training on a task, thus making learning easier. This approach does not necessarily need to solve the task immediately; rather it will improve the chance of coordinating and receiving a reward sufficiently so that the coordinating behaviour may be reinforced.

We show CL to be a feasible solution to the causality dilemma present in tightly coupled MAS. We provide novel empirical evaluation of CL in MAS, with teams of up to 30 agents learning to coordinate using the full continuous state-action space. We show that CL can learn good policies in tightly coupled MAS, including in environments where, without CL, agents fail to learn at all. Additionally, we find that reward shaping (e.g. difference rewards) can be combined with CL to further improve learning performance.

2 BACKGROUND

Transfer learning (TL) is a family of techniques that aims to improve convergence speed and/or converged performance in an unseen problem, through the transfer of knowledge from models that are previously trained on a related problem [8, 9]. Curriculum learning (CL) uses TL to improve learning on a target problem by learning first on an easier training task and progressively increasing the difficulty of the training problem, transferring knowledge along the way [3, 4]. A curriculum is the sequence of training tasks along with rules governing when to switch between tasks.

In multiagent reinforcement learning, teams of agents interact with an environment and each agent tries to maximize the reward it receives from the environment. A global reward G is set to the multiagent teams's performance, but this reward can be noisy as G reflects the actions of all agents and therefore does not properly isolate the contribution of the agent. The reward noise due to other agents have a negative impact on learning [1].

Reward shaping aims to improve the quality of learning by using domain knowledge to make the reward signal more informative. The difference reward D is a shaped reward that better signals the utility of the evaluated agent's sole contribution to the multiagent system. D is the difference between actual G and G with some counterfactual replacement of the evaluated agent [1]. Formally:

$$D_i = G(z) - G(z_{-i} \cup c_i) \qquad (1)$$

where D_i is the agent i's difference reward, z is the collective state-action, or sequence of state-actions for all agents, G is the team performance, z_{-i} is the system state-action or sequence of state-actions for all agents excluding the evaluated agent i, and c_i is a counterfactual state-action or sequence of state-actions that replaces those of agent i.

Cooperative coevolutionary algorithms (CCEAs) extend evolutionary algorithms to multiagent domains by evolving agents' policies independently [6]. Evolutionary algorithms (EAs) are a class of optimization algorithms inspired by evolution that improve some metric for a sequence of values called a genome, with each value in the genome being a gene. A genome is applied to a problem

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

and then evaluated, receiving a fitness score that measures its quality as a solution. An evolutionary algorithm updates a population of genomes using evolutionary operators: selection, recombination, mutation and reinsertion [5]. The expected average fitness score of the population generally increases as evaluations followed by operations are repeated in batches over many generations.

3 CURRICULUM LEARNING FOR THE TIGHTLY-COUPLED MULTI-ROVER DOMAIN

The tightly coupled multi-rover domain [7] is a two-dimensional domain with continuous states and actions. In this domain, multiple rovers with limited sensing capabilities are tasked with interacting with various points of interests (POIs). Each POI has a value that determines the reward for interacting with it. The performance of the multiagent system is the sum of the values of the POIs that have been interacted with during the run. Multiple rovers must simultaneously be within a POI's interaction radius to interact with that POI. The number of rovers a task required (n_{req}) for interacting with a POI in this domain represents the degree of coupling for that task.

In our experiments, we setup the position of the POIs along the edges or corners of a perimeter with a width and height set to some *setup size*. Rovers are initialized in the center of the of this perimeter in an area with a width and height that is 10% of the setup size.

The team performance P is calculated as:

$$P = \sum_{k \in K} \frac{V_k}{\mathcal{D}(k)} \tag{2}$$

where V_k is the value of POI k within the set of POIs K, $\mathcal{D}(k)$ is the closest (i.e. minimum distance metric) any rover gets to POI k during interaction with POI k throughout the entire run.

In this paper, the curricula exclusively change either one of two properties of the domain: a) the *coupling requirement*: the number of rovers required to be within a POI's interaction radius to interact with that POI and b) the *setup size*: the scaling factor for the initial positions of the rovers and POIs. We call the curriculum that modifies the coupling requirement the *coupling curriculum* (Coup Curr). We call the curriculum that modifies the setup size the *size curriculum* (Size Curr). We hand-generate curricula for each tasks For the task with 30 agents, 8 POI, a setup size of 50 and coupling requirement of 6, the coupling curriculum sets the training coupling requirement to 1 for 250 generations, then 2 for 250 generations, then 3 for 500 generations, then 4 for 500 generations, then 5 for 500 generations, then 6 for the remainder of the trial. For the same task, the size curriculum sets the training setup size to 10 for 500 generations, 20 for 500 generations, 30 for 500 generations, 40 for 500 generations, then 50 for the remainder of the trial.

4 EXPERIMENTAL RESULTS

Performances are reported with 95 percent confidence intervals. The performance of agents trained with just G and D (with no suffixes shown in fig. 1) do not use curricula.

The performance curves for the target task with 30 agents, 8 POI and n_{req} =6 are shown in fig. 1. Performance is averaged over 50 statistical trails. A population size of 50 ANNs per agent is used for the CCEA. Training with D and coupling curriculum achieves a performance 14.0 ± 0.84. Training with G and coupling curriculum achieves a performance of 9 ± 1.0. Training with D and size curriculum achieves a performance of 14 ± 1.0. Training with G and size curriculum achieves a performance of 10 ± 1.1. Training with D without either curriculum achieves a performance of 0 ± 0.0. Training with G without either curriculum achieves a performance of 0 ± 0.0.

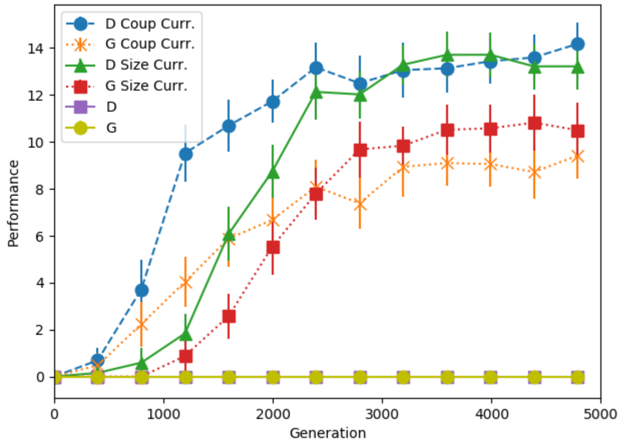

Figure 1: Average multiagent system performance curves using different reward-shaping-curriculum combinations evaluated on target task with 30 agents, 8 POI and n_{req} =6

5 CONCLUSION AND FUTURE WORK

In tightly coupled cooperative tasks, a team of agents must together be in the right place and select the right action at the right time to achieve their goal and receive a reward. The failure of enough team members to coordinate their behavior may make it difficult for other agents to determine the utility of their own actions. Domains with tight coupling present a causality dilemma; one agent can not learn unless other agents are doing the right thing, and other agents will not do the right thing without learning. We evaluate the effect of training with CL on the tightly coupled multi-rover domain. Without using CL, agents were not able to learn to coordinate. Incorporating D in addition to CL further increased converged performance.

We show that CL allows agents to learn good policies in tightly coupled MAS, including in environments where, without CL, agents fail to learn. Additionally, we demonstrate that CL can be combined with shaped multiagent reward signals, such as difference rewards, to further improve learning performance.

ACKNOWLEDGMENTS

This work was partially supported by the National Aeronautics and Space Administration under Grant No. 80NSSC18K0941. PM's visit to OSU was funded by a Fulbright-TechImpact Award.

REFERENCES

[1] Adrian Agogino and Kagan Tumer. 2004. Efficient evaluation functions for multi-rover systems. In *Genetic and Evolutionary Computation Conference*. Springer, 1–11.

[2] Richard Bellman. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

[4] Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone. 2016. Source task creation for curriculum learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 566–574.

[5] Hartmut Pohlheim. 2003. Evolutionary algorithms. *GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB Documentation. GEATbx* (2003).

[6] Mitchell A Potter and Kenneth A De Jong. 1995. Evolving neural networks with collaborative species. In *Summer Computer Simulation Conference*. SOCIETY FOR COMPUTER SIMULATION, ETC, 340–345.

[7] A. Rahmattalabi, J. J. Chung, M. Colby, and K. Tumer. 2016. D++: Structural credit assignment in tightly coupled multiagent domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4424–4429.

[8] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

[9] Lisa Torrey and Jude Shavlik. 2010. Transfer Learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 242–264.