Learning Behaviors from a Single Video Demonstration Using Human Feedback

Extended Abstract

Sunil Gandhi, Tim Oates, Tinoosh Mohsenin
University of Maryland Baltimore County
Baltimore, MD
{sunilga1,oates,tinoosh}@umbc.edu

Nicholas R. Waytowich
US Army Research Laboratory
Aberdeen, MD
nicholas.r.waytowich.civ@mail.mil

ABSTRACT

In this paper we present a method for learning from video demonstrations by using human feedback to construct a mapping between the internal state representation of the agent and the visual representation from the video. In this way, we leverage the advantages of both these representations, i.e., we learn the policy using agent centered state representations, but are able to specify the expected behavior using video demonstrations. We show the effectiveness of our method by teaching a hopper agent in the MuJoCo simulator to perform a backflip using a single video demonstration generated in MuJoCo as well as from a real-world YouTube video of a person performing a backflip.

KEYWORDS

Deep learning; Reinforcement learning, Human-Robot Interaction

ACM Reference Format:
Sunil Gandhi, Tim Oates, Tinoosh Mohsenin and Nicholas R. Waytowich. 2019. Learning Behaviors from a Single Video Demonstration Using Human Feedback. 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

A common way to train a reinforcement learning agents is by optimizing the policy to maximize a well-specified reward function [7, 10, 13]. But, even for experts, designing reward functions is a complex and time-consuming process, and for some tasks, they are too difficult to be specified by hand [3]. Consequently, other approaches for teaching new skills to autonomous agents have been explored. A common method for teaching new skills is through example demonstrations [1, 2, 8]. Often, these demonstrations are collected in the form of lower dimensional representations like angles of joints of a robotic arm (often referred to as the standard representation). However, collection of demonstrations using the standard representation is a difficult process and requires expertise. Other approaches like [5, 6, 9, 11, 12, 14] learn the expected behavior by observing the task being performed in a video recording. This method of providing expected behavior is simple, intuitive and can leverage a large number of videos on the web. However, they do not allow for any corrective feedback to improve the performance of the agent. There are other methods like the ones proposed in [3, 4, 15, 16] that learn the task using human feedback. However, these methods do not use the simple and intuitive method of showing the expected behavior through videos.

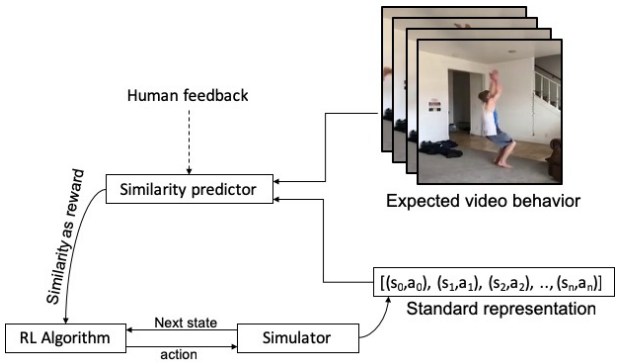

Figure 1: Schematic illustration of our architecture

We believe that a combination of these approaches could result in a powerful method for teaching autonomous agents to perform a task. In this paper, we leverage both the standard representations and visual representations by learning a mapping between the two. This simplifies the optimization of the reinforcement learning algorithm as we can learn directly on the lower dimensional standard representation but are still able to easily specify the desired behavior using video demonstrations. To achieve this, we utilize human feedback to learn similarity predictor between the standard and visual representations. Learning this mapping between standard and visual representations is a difficult problem for autonomous agents, especially if the number of example behaviors is limited. However, this can be easily performed by humans. In this paper, we ask human participants to rate the similarity between behaviors produced by the learning agent alongside example video clips of the desired behavior. We show that human feedback enables our agent to learn complex behaviors from a single video demonstration.

2 ARCHITECTURE

The goal of our framework is to teach a reinforcement learning agent to replicate the behavior shown in a video demonstration. To train this agent, we use a single video demonstration of the task and human feedback that indicates the similarity between the agent's behavior and the video demonstration.

The architecture of our system is illustrated in Figure 1. Our method consists of three processes running asynchronously. First,

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.

the similarity function S is learned by training a deep neural network using human feedback collected in the form of similarity ratings between the demonstrated clip and the corresponding behavior of the learning agent. This network enables the comparison between any agent-generated trajectory to that of expected behavior provided by the demonstration video. Second, the comparison between the agent's behavior and the expected behavior using the similarity network is used as a reward signal for optimizing a reinforcement learning algorithm. During the optimization, the RL agent generates trajectories that are similar to the video demonstration with respect to the similarity function S. Finally, segments of the trajectories generated during the optimization of the reinforcement learning algorithm are provided to a human for feedback. These three processes, i.e., training of the similarity network, optimization using reinforcement learning and collecting human feedback run simultaneously and asynchronously. Under this framework, any traditional reinforcement learning algorithms can be used in principle, however, we use trust region policy optimization (TRPO) as it works well when the reward function is not stationary [10].

To learn the similarity predictor S, we use human feedback to provide ratings of how similar the agent's behavior is to the video. We show two short clips to the human. One clip is from the video demonstration and another is the corresponding imitation by the RL agent. The video demonstration and imitation clips are 0.3 seconds long. The human provides feedback as a rating indicating the similarity between both clips ranging from 1 to 5, where a rating of 1 indicates that behaviors in both clips are not at all similar, and 5 indicates that they are very similar. We collect the feedback in two stages. First, during a pre-training stage, feedback is collected with the agent performing random actions. Later during training, we use trajectories generated by the RL agent for getting human feedback to further refine the similarity predictor.

3 EXPERIMENTS

We teach the hopper robot to perform a backflip in the MuJoCo environment from a single video demonstration and human feedback. We compare our method to a traditional reinforcement learning algorithm using a hand-coded reward function as well as the learning from human preferences method [3].

For the traditional reinforcement learning comparison, we used a reward function from [3] that was constructed to get a hopper agent to backflip, where the more backflips the agent achieves in a given period of time, the higher reward it receives. We use this "backflip reward function" to train a traditional reinforcement learning agent using TRPO. We save the video corresponding to the trajectory of the TRPO agent with the maximum backflip reward. We use this video as the demonstration video for teaching our agent to replicate the backflip.

We also compare our method to **learning from human preferences** by [3] that learns the reward function by collecting human preferences between pairs of trajectory segments. To learn from human preferences, we collect preferences with the goal of performing as many backflips as possible in 8 seconds.

For both our method and [3], we collected 350 total annotations. 200 of those annotations were collected during the pretraining stage and 150 annotations using rollouts from the policy network that is being optimized asynchronously.

Algorithm	# of Human samples	Maximum backflip reward	Total backflips
TRPO	n/a	19859.81	2
[3]	350	2064.27	1
Our method	350	3365.18	2

Table 1: Performance comparison on the Backflipping task

We used two metrics to compare our method's performance: 1) total reward achieved from the backflip reward function and 2) total number of backflips performed during an episode (counted manually from the produced trajectory). Table 1 shows the maximum backflip reward and corresponding number of backflips performed in 8 seconds. Not surprisingly, the traditional TRPO agent achieves the highest reward as it was directly optimizing over the backflip reward function. Even still, our method, which does not require a reward function to optimize over, achieves higher reward compared to the learning from preferences method. Even though our method does not achieve the same amount of reward as the TRPO baseline, our agent performs same number of backflips as that of the TRPO baseline and one more backflip than the preferences method achieved. We can get comparable performance to the TRPO baseline without requiring the use of a handcrafted reward function, and can achieve significantly better performance compared to the human preference method using the same amount of human feedback. Additionally, our method has the potential to learn any task by utilizing the vast number of videos on the web.

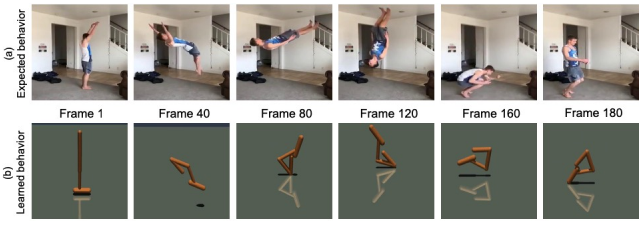

Figure 2: Comparison of the backflip in the real-world video demonstration and by our agent

We also demonstrate the ability of our method to learn from an actual YouTube video. We use the optimized similarity predictor to teach our hopper agent to backflip using a real-world video from YouTube of a human performing a backflip. The frames of the video are shown in figure 2(a). It can be easily observed from Figure 2 that since the kinematics of the human and hopper robot are very different, the mapping between between them is ambiguous. This makes both the annotation and learning of the behavior a challenging problem. We collected 200 annotations during a pre-training stage and 165 annotations while the reinforcement learning policy was being optimized. Figure 2(b) shows the learned backflip performed by the RL agent trained using our method. This shows the effectiveness of our method in replicating behavior demonstrated in single, non-annotated YouTube video.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 1.

[2] Aude Billard and Daniel Grollman. 2013. Robot learning by demonstration. *Scholarpedia* 8, 12 (2013), 3824.

[3] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*. 4299–4307.

[4] W Bradley Knox and Peter Stone. 2008. Tamer: Training an agent manually via evaluative reinforcement. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*. IEEE, 292–297.

[5] Jangwon Lee and Michael S Ryoo. 2017. Learning robot activities from first-person human videos using convolutional future regression. *Image* 500 (2017), 500.

[6] Yuxuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018. *Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation*. Technical Report. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-37.html

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[8] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *Icml*. 663–670.

[9] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018. SFV: Reinforcement Learning of Physical Skills from Videos. *ACM Trans. Graph.* 37, 6, Article 178 (Nov. 2018), 14 pages.

[10] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International Conference on Machine Learning*. 1889–1897.

[11] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. 2017. Time-Contrastive Networks: Self-Supervised Learning from Video. *arXiv preprint arXiv:1704.06888* (2017).

[12] Pierre Sermanet, Kelvin Xu, and Sergey Levine. 2016. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699* (2016).

[13] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529 (2016), 484–503. http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

[14] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. 2017. Third-person imitation learning. *arXiv preprint arXiv:1703.01703* (2017).

[15] Garrett Warnell, Nicholas R. Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. In *AAAI*.

[16] Christian Wirth, Johannes Furnkranz, Gerhard Neumann, et al. 2016. Model-free preference-based reinforcement learning. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*. 2222–2228.