Behavioral Cloning from Observational Reinforcement

Anurag S. Aribandi
School of Computing
University of Utah
anurag.aribandi@utah.edu

Abstract

Feedback and learning from observation are both two key parts of not only learning on a fundamental level but of many classical and state of the art RL algorithms. While Behavioral Cloning from Observation and TAMER address these problems separately, this project is motivated by the opportunity to combine them for a potentially more versatile algorithm. To combine the inverse dynamic model of BCO along with the accessible feedback of TAMER. This project will explore the use of state-only feedback to teach an agent to learn an optimal policy shown by a demonstrator. When implemented in MountainCar and tested in a pilot study, users were found to be able to quantitatively produce a working policy and qualititatively found the process of giving feedback accessible despite having no prior experience.

1 Introduction and Motivation

Learning is one of the foremost research areas explored in Computer Science. Research in Reinforcement Learning (RL) in particular quite evidently partakes in this and a significant number of leading research deals with new ways of teaching artificial agents. Two important aspects of learning in on a fundamental level, and learning in RL are feedback and learning from demonstration. Learning from demonstration can be quite an abstract concept, what exactly constitutes a demonstration? Classic RL work generally defines demonstrations as some form of sequence of state and action transitions. Simply put, given a state that describes the environment or goal of the agent in some way, the demonstration should how represent how the agent acts in accordance with said environment. When these state-action pairs are collected in multitude, they form a trajectory. Many classic learning algorithms involve learning this mapping of states to actions, or a policy. Behavioral Cloning (BC), for example, learns a neural network to output an action given a representation of the agent's state. While these algorithms that treat demonstrations as such have had significant success, learning purely from observation is an integral part of learning. And gathering action information can be computationally intensive if the environment space is particularly complex, or if access to said environment is limited. Behavioral Cloning from Observation[9] (BCO) is an algorithm that attempts to combat this problem. It does so by implementing BC but with the help of an inverse-dynamic model, a model that can infer actions that lead to a given state change. The utilization of this model has been shown to lead to comparable learning performance with BC given state-only demonstrations.

Similarly, while the term may be straightforward, when considered from an RL perspective, the term "feedback" can also be quite abstract. Feedback comes in many forms, and due to this, there are also a multitude of algorithms that deal with different forms of feedback. However, typically feedback implies that the learning algorithm designer, has to implement some type of reward function for the agent that evaluates the agent based on its state with respect to its environment and task. The issue with this is not only is this difficult in general for people who are not domain experts, but it also increases in difficulty to formulate specific feedback the more complex the environment space and

Class Project for CS 6960: Human-AI Alignment at the University of Utah, Fall 2022.

task get. The TAMER[5] framework introduces the key idea of learning a human's reinforcement model for feedback as a policy instead of relying on demonstrations. The feedback in question is in the form of binary "yes" or "no" signals. The simplicity of the feedback involved makes it accessible to train an agent in this way as evaluating an agents performance is easier than showing it what to do.

Both of the described algorithms address the problems they aim to solve in an interesting way, this project is motivated by attempting to combine the strengths and novelty of both of them. This report proposes a novel algorithm, Behavioral Cloning from Observational Reinforcement (BCOR) which involves using an inverse-dynamic model to interpret state-only feedback from a human. This work is motivated by the opportunity to combine them for a potentially more versatile algorithm. When tested on MountainCar, BCOR was found to quantitatively improve on suboptimal demonstrations after the algorithms reinforcement phase. The report elaborates on related work in section 2, followed by a description of the implementation of the algorithm in section 3, and concludes with a brief discussion on the experimental design and results and future potential and limitations of the work done.

2 Related Work

The related work will be broadly split into two perspectives : different types of feedback implemented in learning, and dealing with suboptimal demonstrations.

2.1 Feedback

Chisari et al[1] introduced the CEILING framework which Provides feedback primarily in the form of evaluative feedback (while the initial learned policy is executed) similar to TAMER, if the trajectory can be easily corrected, then feedback in the form of actions (corrective) after observation of the agent's state from images is given. As touched upon in the earlier section, Knox et al. developed the TAMER[5] framework where the agent learns a policy that selects an action that maximizes a reinforcement models estimate that is learned from a human reinforcing it with binary signals (yes or no). Rather than simply aiming to add expert labels to improve a novice policy, Kelly et al.[4] iterated on DAgger and added the expert labels only when the agent enters unsafe regions. Once the agent exits the unsafe regions, control is handed back to the agent. In HGDAgger, feedback is interpreted primarily from a safety perspective rather than solely with the aim of performance improvement. DemPref[8] combines learning from feedback and preferential learning first using the users demonstrations to learn a prior over reward functions. This then helps ground and optimize the amount of times the human in the loop is queried for preferences. The human then ranks a demonstration trajectory and two generated trajectories as the form of feedback. The chosen trajectory replaces the demonstrated trajectory and the process repeats until convergence. Jain et al.[3] Presents a co-active online learning framework where the human teacher iteratively provides improvements over the trajectory currently being taken. This work is Similar to preference learning in that it involves preferentially giving better trajectories. However it is an iterative improvement, the way the trajectories are improved can give information regarding rewards. Is especially relevant for the real world evaluation where subtle changes in object orientation may drastically change the quality of a trajectory. CyberSteer[2] also uses feedback in the form of human given actions. First, the agent is trained to distinguish between human and non-human actions. The reward is calculated based on the likelihood that the action taken is similar to the action executed by a human. The magnitude of feedback is based on how close the human is to the agent's action. Watkins et al.[11] developed a model that learns to interpret advice is first learned. The way the advice distillation is trained is through a two step process to first interpret simple advice and then more complex advice. That advice is subsequently used as feedback to inform the agent on what actions to take, goals to move towards or sub-tasks to complete. The agent is finally evaluated without additional feedback from the human. The work done on COACH[6] offers an interesting perspective on feedback where they consider feedback to be policy dependant. Trainers are taught to provide diminishing returns(gradual decreases in positive feedback for good actions as the agent adopts those actions), differential feedback(varied magnitude of feedbacks depending on the degree of improvement or deterioration in behavior), and policy shaping(positive feedback for suboptimal actions that improve behavior and then negative feedback after the improvement has been made), all of which are policy dependent.

2.2 Suboptimal Demonstrations

Less work in comparison was examined in this aspect just to get a brief impression of previous attempts at mitigating suboptimal demonstrations. Wu et al.[12] Used demonstrations as reward shaping potential to generate state and action pairs from a GAN. They Interpret demonstrations as advice rather than something to explicitly imitate to account for assumed sub-optimality on the demonstrators part. Wang et al.[10] use lower bound reward constraints which helps in learning human-desired policies even when supplied with suboptimal demonstrations. Q-filtering is a method developed by Nair et al.[7] whereby they only keep the terms of the behavioral cloning loss for which the demonstrated action has higher Q value than the action returned by the policy.

While there has evidently been a multitude of work that involves a variety of feedback types and methods of interpreting feedback, I did not come across any prior work that specifically aims to interpret state-only feedback with the help of an inverse-dynamic model.

3 Proposal and Methodology

The proposed method to combine the two algorithms is to use an inverse dynamic model to interpret state-only feedback. This is an attempt to combine what I believe to be the strengths of both TAMER and BCO. I hope increase the dimensionality of feedback available to users and also make the inverse dynamic BCO process more accessible to non-experts. I theorize that this will also be a viable method to obtain functioning policies from suboptimal demonstrations. The broad overview of the algorithmic process is depicted in figure 1.

The algorithm was implemented in the MountainCar environment where the agent is tasked with getting out of the valley to the flag within 200 time steps. The agent can accelerate left or right or not at all and receives a reward of -1 each tilmestep that it does not reach the goal. Similar to BCO, first random interaction data is collected and formatted to be used to train the inverse dynamic model. In this case, the inverse dynamic model was implemented as a neural network that takes a state and a future state (described by a tuple of the cars current x-position and velocity) and outputs the most likely action. After evaluating the model, the demonstration is collected from the human where the actions are mapped to the arrow keys (the number of demonstrations given can be fine tuned). The resultant state-only trajectories are collected, formatted to be used as inputs for the inverse dynamic model and the resultant actions are combined with the original trajectories to be used for BC. The policy is then evaluated and the human supervisor or trainer has two options. If they see the agent executing a suboptimal policy (by their own subjective judgement), they may choose to reinforce the agent by suggesting a state to indicate feedback. In the context of the implementation in the MountainCar environment, this involves clicking the mouse in the game window. Once this event is recorded, the mouse cursor's position is extrapolated and scaled to the environments x-position attribute. This is then appended with the agent's current state as a next state to use as a "goal" and is fed into the inverse dynamic model to produce the recommended action. This action is then taken and the process is looped by updating the agent's current position and fetching new recommended actions. This is done until the agent comes within a certain proximity of the state suggested by the human. Control is then handed over back to the agent and the human can choose to reinforce it again or not for the remainder of the episode length (200 time steps).

The whole reinforcement process occurs for the duration of the MountainCar episode. If the human deems the learned policy through BCO to be adequate and does nothing for the episode, then the learned policy will remain unchanged and will be evaluated. Before the episode is over, if the human decides to provide feedback, then the reinforcement process will start. Once the episode is done, the state-action pairs collected during the reinforcement process are used to retrain the policy which is subsequently evaluated.

4 Experiments

The main points I wanted to explore within the scope of this project were its potential use to improve performance of policies learnt on suboptimal demonstrations, and its accessibility to non-experts. A

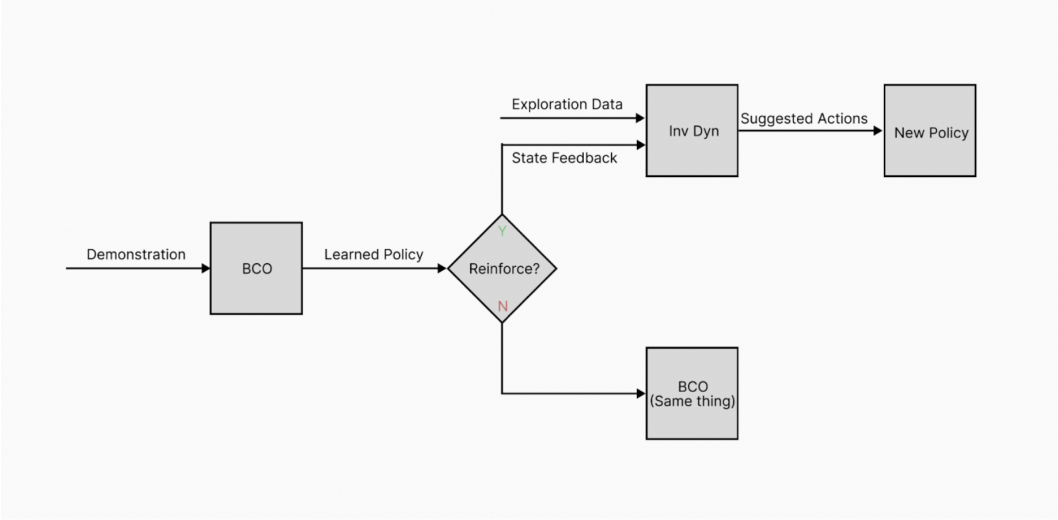

Figure 1: Illustration of BCOR

short quantitative pilot was conducted with two users. Each user was provided with a suboptimal demonstration where the agent was taught to rock about the valley it is meant to get out of. User 1 was able to achieve a score of -131 and User 2 was able to achieve a score of -156 on their first try (which is by definition better than the redundant demonstration of simply rocking about). They were explained the goal and given an idea on how to get out of the valley and then given control of the reinforcement phase. However, neither of the users had experiencing controlling MountainCar or had seen the environment before. Users were also asked subjectively about their opinions on giving feedback to alter the performance of the bot in this manner. User 1 stated that it was intuitive while User 2 said that while it was easy to get the hang of, felt that the time constraints made it possible to mess up and felt they could have done better given multiple tries. Users were able to objectively correct a bad demonstration on their first time and also subjectively felt it was accessible enough even though they had no prior experience with it. It is important to note however, that these are mere preliminary results in a simplified environment and there was possible bias at play in the users subjective responses.

5 Future Work and Limitations

The immediate future potential of this work is to conduct more elaborate quantitative and qualitative analysis to further prove this frameworks viability in potentially more complicated environment spaces. MountainCar is perhaps one of the simplest environments to test this framework in. Implementing this algorithm in a more complicated state and action space would more confidently speak to it's viability in more complex spaces. While this framework was partly inspired by TAMER, this implementation does not actually explicitly learn how or why a reinforcement is given. An interesting avenue to explore could involve an experimental reinforcement phase where an intentionally suboptimal demonstration or series of demonstrations are given and a reinforcement model is learned after observing how the human differently reinforces different demonstrations. This could help automate the reinforcement process when given the actual demonstrations in practice and help reduce the decision making strain on the human. This leads into the potential problem of the strain the reinforcement process could potentially have on the reinforcer. Even when implemented in MountainCar, giving the reinforcement feedback to the agent is something that, while accessible, may not be trivial to do perfectly especially when given the time constraints of the episode. Considering that a big motivation of this work is to improve accessibility of providing feedback, it is important to keep in mind the potential burden of providing such feedback if given in tighter time constraints or complex task spaces.

Although limited in scope, this project was able to demonstrate the proof of concept of this framework. The conducted experiments were able to show the assistance in correcting suboptimal demonstrations in the chosen domain. This project also aimed to showcase the potential of using inverse dynamic models in general for interpreting feedback. For example, a future project avenue could involve using this model to assist in interpreting areas of danger and learning safer policies for an agent when signalled states to avoid. The proposed framework could also be integrated with other learning methods, not just learning from demonstrations. I hopefully anticipate that this framework represents another way of interpreting and utilizing feedback in a novel way that can contribute to furthering research in the field of teaching and RL.

References

[1] Eugenio Chisari, Tim Welschehold, Joschka Boedecker, Wolfram Burgard, and Abhinav Valada. Correct me if i am wrong: Interactive learning for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3695–3702, 2022.

[2] Vinicius G Goecks, Gregory M Gremillion, Hannah C Lehman, and William D Nothwang. Cyber-human approach for learning human intention and shape robotic behavior based on task demonstration. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018.

[3] Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning trajectory preferences for manipulators via iterative improvement. *Advances in neural information processing systems*, 26, 2013.

[4] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[5] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

[6] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, pages 2285–2294. PMLR, 2017.

[7] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[8] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. *arXiv preprint arXiv:1906.08928*, 2019.

[9] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[10] Zhaorong Wang, Meng Wang, Jingqi Zhang, Yingfeng Chen, and Chongjie Zhang. Reward-constrained behavior cloning. In *IJCAI*, pages 3169–3175, 2021.

[11] Olivia Watkins, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Jacob Andreas. Teachable reinforcement learning via advice distillation. *Advances in Neural Information Processing Systems*, 34:6920–6933, 2021.

[12] Yuchen Wu, Melissa Mozifian, and Florian Shkurti. Shaping rewards for reinforcement learning with imperfect demonstrations using generative models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6628–6634. IEEE, 2021.