Leveraging Sub-Optimal Data for Human-in-the-Loop Reinforcement Learning

Extended Abstract

Calarina Muslimani
University of Alberta
Edmonton, Canada
muslima@ualberta.ca

Matthew E. Taylor
University of Alberta
Alberta Machine Intelligence Institute (Amii)
Edmonton, Canada
matthew.e.taylor@ualberta.ca

ABSTRACT

To create useful reinforcement learning (RL) agents, step zero is to design a suitable reward function that captures the nuances of the task. However, reward engineering can be a difficult and time-consuming process. Instead, human-in-the-loop (HitL) RL approaches allow agents to learn reward functions from human feedback. Despite recent successes, many of the HitL RL methods still require numerous human interactions to learn successful reward functions. To that end, this work introduces Sub-optimal Data Pre-training, SDP, a method that leverages reward-free, sub-optimal data to improve the feedback efficiency of HitL RL algorithms. We demonstrate that SDP can significantly improve over state-of-the-art HitL RL algorithms in three DMControl environments.

KEYWORDS

Reinforcement Learning; Human-in-the-Loop; Preference Learning

ACM Reference Format:
Calarina Muslimani and Matthew E. Taylor. 2024. Leveraging Sub-Optimal Data for Human-in-the-Loop Reinforcement Learning : Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

In reinforcement learning (RL), an agent's goal is to interact with an environment in order to maximize its total reward [14]. It is assumed that the environment provides an agent with a well-defined reward function that captures all task complexities. But where does this reward function actually come from? Reward functions are hand-engineered by humans in what often can be a tedious and non-trivial pursuit [1]. As the complexity of tasks increases, so does the time and effort required to design a suitable reward function. Further, there have been notable examples of reward misspecification, in which RL agents discovered and exploited unintended shortcuts in the reward function [5, 13].

A promising alternative is to learn reward functions directly from human feedback. In this paradigm, humans can provide feedback in the form of preferences or scalar signals, which are then used to learn a reward function that is consistent with human desires [7, 8]. Despite recent progress, existing preference- and scalar-based RL methods still require thousands of human queries to learn an adequate reward function [3, 8]. Prior work attempts to mitigate this issue through several mechanisms, including active learning [8, 11], data augmentation [10, 12], semi-supervised learning [10, 18], and meta-learning [6, 9]. Alternatively, this work takes inspiration from offline RL. Yu et al. [17] found that in settings where there is an abundance of unlabeled (i.e., reward-free), low-quality data, one way to use this data in offline RL is labeling all such data with a reward of zero (i.e., the minimum task reward). As low-quality data is arguably the easiest type of data to obtain, we present Sub-optimal Data Pre-training, *SDP*, an approach that leverages abundant sub-optimal, unlabeled data to improve learning in HitL RL methods.

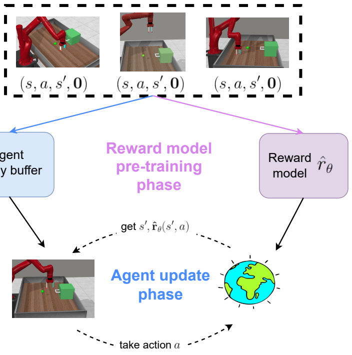

Figure 1: Overview of SDP: After pseudo-labeling all sub-optimal data with rewards of zero, we then pre-train our reward model with this data set. During the agent update phase, we initialize our RL agent's replay buffer with the same pseudo-labeled data set. We then interact in the environment and make learning updates.

2 REWARD LEARNING FROM HUMAN FEEDBACK

This paper assumes that we are in an MDP/R setting (i.e., reward-free), where our goal is to learn a good policy while also learning a reward function from human feedback. In preference-based learning, two segments, σ^0 and σ^1, are compared by a teacher. If the teacher preferred segment σ^1 over segment σ^0, then the target y is

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

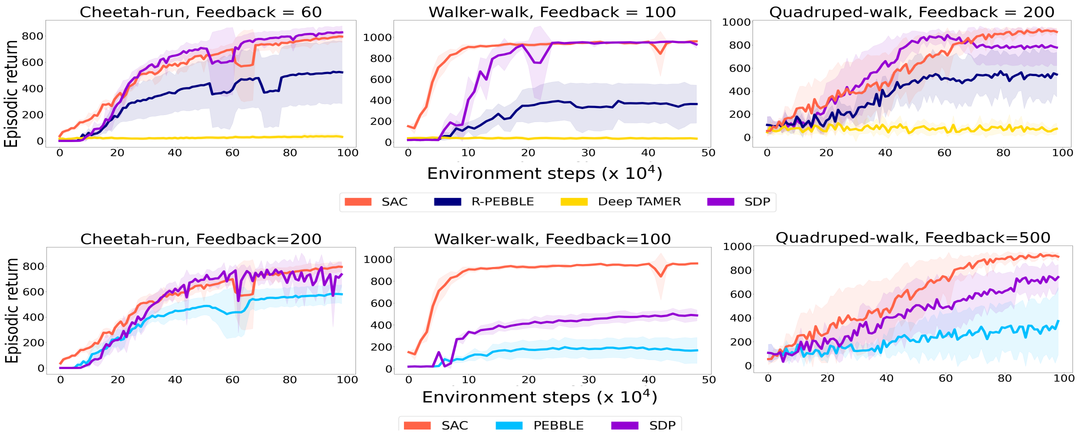

Figure 2: Top: Scalar feedback experiments. Bottom: Preference learning experiments.

set to 1, and if the converse is true y is set to 0. If both segments are equally preferred, then y is set to 0.5. As feedback is collected, it is stored as tuples (σ^0, σ^1, y) in the data set D_{RM}. Then, we follow the Bradley-Terry model [2] to define a preference predictor using the reward function $\hat{r}_\theta$. Intuitively, this model assumes that the probability of the teacher preferring a segment depends exponentially on the total sum of predicted rewards along the segment. To train the reward function, we update $\hat{r}_\theta$ by minimizing the standard binary cross-entropy objective. In the scalar feedback setting, we can apply standard regression as the teacher assigns numerical ratings to trajectory segments instead of preferences.

3 SUB-OPTIMAL DATA PRE-TRAINING

SDP comprises two phases: (1) the reward model pre-training phase and (2) the agent update phase. In the reward model pre-training phase, we obtain 50,000 (state, action) transitions, σ_{sub}, from a random policy and pseudo-label them with a reward of 0. This data set $D_{\text{sub}} = \{(\sigma_{sub}, 0)^i\}_{i=1}^N$ is used to optimize the reward model $\hat{r}_\theta$ using the standard mean squared loss. As a result, the reward model $\hat{r}_\theta$ becomes pessimistic because it learns to associate all sub-optimal transitions with a low reward. Without such a prior, the reward model would initially have random estimates for the sub-optimal transitions. The only way to improve such estimates would be to obtain actual feedback from a teacher. Therefore, by pseudo-labeling the sub-optimal transitions with 0, we obtain free supervision to give our reward model a helpful bias.

Next, in the agent update phase, we initialize the RL agent's replay buffer with D_{sub}. The RL agent then briefly interacts with its environment and performs gradient updates according to its loss functions. This component is necessary as it changes the RL agent's policy and generates new transitions, which are then stored in both the agent's replay buffer and the reward model's replay buffer, D_{RM}. In standard scalar- and preference-based HitL RL approaches, we query the teacher for feedback on trajectory segments sampled from D_{RM}. Therefore, when it is time for the teacher to provide their first set of feedback, the feedback can cover a different region of the state and action space, relative to the original sub-optimal data (as D_{RM} was empty prior to the agent update phase). After these phases, any off-the-shelf HitL RL algorithm can proceed.

4 RESULTS

We apply SDP to both scalar- and preference-based RL approaches. We benchmark it against four algorithms: PEBBLE [8] for a preference-based RL comparison, Deep TAMER [16] for a scalar-based RL comparison, R-PEBBLE (a regression variant of PEBBLE) as another scalar-based RL benchmark, and SAC [4]. SAC is an oracle baseline as it learns while accessing the true reward function, which is unavailable to the other algorithms. To evaluate performance, we use a scripted teacher that provides either a scalar rating of a single trajectory or preferences between two trajectories according to the oracle reward function. We train all algorithms for one million time steps. We evaluate SDP in three DMControl environments [15] — Walker-walk, Cheetah-run, and Quadruped-walk. All results are averaged over five seeds with shaded regions indicating a 95% confidence interval. To test for significant differences in performance (e.g., final performance and area under the curve, AUC), we perform a Welch t-test with a p-value of 0.05. Figure 2 shows the resultant performance for the scalar-based feedback experiments (top figure) and preference-based feedback experiments (bottom figure). In all experiments, we found that SDP significantly improves ($p < 0.05$) over the state-of-the-art HitL RL algorithms in either learning efficiency (i.e., AUC) or final performance. The simplicity and effectiveness of SDP allow for it to be easily combined with off-the-shelf HitL RL algorithms to improve learning. Overall, this work takes an important step toward considering how HitL RL approaches can take advantage of readily available sub-optimal data.

ACKNOWLEDGMENTS

Part of this work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Digital Research Alliance of Canada; Huawei; Mitacs; and NSERC.

ETHICS STATEMENT

The goal of preference and scalar-based RL is to learn reward functions that align with a teacher's preferences or desires. However, there is the possibility that a hostile user (i.e., with harmful desires) provides feedback to teach the RL agent negative behaviors. Therefore, although it is important to improve the performance and efficiency of these methods, it is equally important for further research to focus on the development of safe HitL RL approaches that can safeguard against malicious outcomes.

REFERENCES

[1] Serena Booth, W. Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. 2023. The Perils of Trial-and-Error Reward Design: Misdesign through Overfitting and Invalid Task Specifications. *Proceedings of the AAAI Conference on Artificial Intelligence*.

[2] Ralph Allan Bradley and Milton E. Terry. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* 39, 3/4, 324–345. http://www.jstor.org/stable/2334029

[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*.

[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*.

[5] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. 2017. Inverse Reward Design. *Advances in Neural Information Processing Systems*.

[6] Donald Joseph Hejna III and Dorsa Sadigh. 2023. Few-Shot Preference Learning for Human-in-the-Loop RL. In *Conference on Robot Learning*.

[7] W Bradley Knox and Peter Stone. 2009. Interactively Shaping Agents via Human Reinforcement: The TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*.

[8] Kimin Lee, Laura Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *International Conference on Machine Learning*.

[9] Calarina Muslimani, Alex Lewandowski, Dale Schuurmans, Matthew E Taylor, and Jun Luo. 2022. Reinforcement Teaching. *Transactions on Machine Learning Research* (2022).

[10] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2022. SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. In *International Conference on Learning Representations*.

[11] Burr Settles. 2009. Active learning Literature Survey. (2009).

[12] Connor Shorten and Taghi M Khoshgoftaar. 2019. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* (2019).

[13] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and Characterizing Reward Gaming. *Advances in Neural Information Processing Systems*.

[14] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An Introduction*. MIT press.

[15] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

[16] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[17] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. 2022. How to Leverage Unlabeled Data in Offline Reinforcement Learning. In *International Conference on Machine Learning*.

[18] Xiaojin Jerry Zhu. 2005. Semi-supervised Learning Literature Survey. (2005).