Decreasing the number of demonstrations required for Inverse Reinforcement Learning by integrating human feedback

Zanyar Ogurlu [1], Supervisors: Luciano Cavalcante Siebert [1], Antonio Mone [1]

[1]TU Delft, The Netherlands

Background

- *Reinforcement learning:* Teaching an agent how to optimally make sequential decisions within a certain context.
- *Problem:* We often have an *unknown reward function*. It might be simple to define reward weights in games like chess, but much more difficult in more complicated settings.
- *Inverse Reinforcement Learning or IRL* (learning from demonstrations) and *Reinforcement Learning from Human Feedback or RLHF* (learning from feedback) help us learn the reward function. Figure 1 shows the results of an example reward learning process using IRL.
- *Adversarial IRL:* The inverse reinforcement learning algorithm that was used during this study. It was picked specifically because it is good at resolving two kinds of ambiguities: Picking the optimal policy from demonstrations and picking the optimal reward function from the policy.
- *Preference comparisons:* The agent performing RLHF will be making use of preference comparisons, which is simply the comparison between two trajectory segments - which one is better?

Figure 1. Learning reward function via IRL

Research Question

To what extent can RLHF complement IRL to reduce the number of expert demonstrations needed for IRL?

Subquestions:

- How can expert feedback be incorporated into a system that uses Adversarial IRL and what kind of advantages does this approach bring?
- In which circumstance does adding RLHF to a system that uses IRL not benefit the performance of the system?

Experimental Setup

- The cartpole environment from the *gymnasium* library will be used. The visualization of this environment is shown in figure 2.
- We chose the cartpole environment since it has a simple and known reward function. This is mostly due to a lack of time and resources, since we are performing RL, IRL and RLHF all at once, our experiments are computationally intensive.

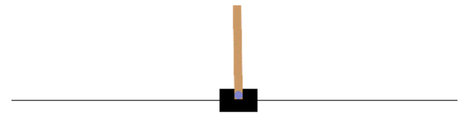


Figure 2. CartPole environment from the *gymnasium* library

Methodology

Figure 3 shows a high-level visualization of our methodology. This is followed by a more detailed step-by-step explanation.

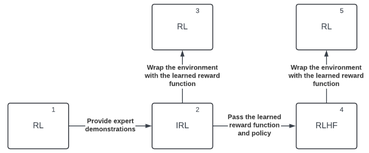
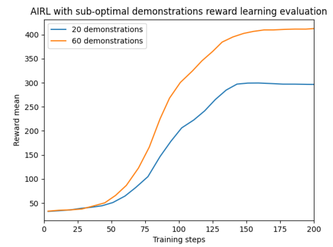


Figure 3. Experiment methodology

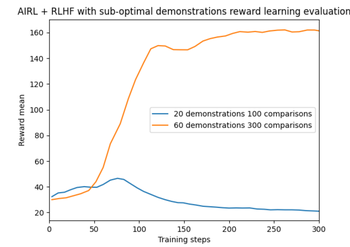
1. Generating sub-optimal expert demonstrations using Deep Q-Learning. The sub-optimality of the demonstrations will be due to an insufficient degree of training on the environment (reward mean: 474, standard deviation: 64).
2. Applying AIRL to the environment, blind to the actual reward function for that environment. The agent will have learned an initial reward function from sub-optimal demonstrations, which we expect to be inadequate.
3. Wrapping the environment with the reward function learned via AIRL, and performing reinforcement learning on this environment. The results of this learning process will be evaluated by resorting to the original reward function for the environment.
4. Fine-tuning the reward function via a preference comparison RLHF where the trainer makes a preference between different trajectory segments.
5. Wrapping the environment with the reward function learned via RLHF, and performing reinforcement learning, to see how well RLHF managed to model the reward function at this stage.
6. Repeating the experiment with varying number of expert demonstrations.

Results and Discussion

We first trained an AIRL model with optimal demonstrations, to verify that in such a setting, AIRL by itself will yield an accurate reward function. Then, we trained an AIRL model with sub-optimal expert demonstrations. We verified that AIRL starts to perform poorly by itself when the provided demonstrations are sub-optimal, especially as the number of demonstrations drop. Finally, we performed a training on the environment via our proposed approach. We saw that our approach did not improve the reward function learned via AIRL, in fact it dropped the accuracy of the reward function learned via AIRL at the beginning. Furthermore, this approach will not help decrease the number of expert demonstrations required by AIRL. The results are shown in figure 4.



(a) Reinforcement learning using AIRL rewards provided sub-optimal demonstrations



(b) Reinforcement learning using AIRL+RLHF rewards provided sub-optimal demonstrations

Figure 4. Experiment results

Conclusion and Limitations

- IRL calculates an accurate reward function when provided optimal demonstrations.
- IRL starts to perform poorly when the provided demonstrations are sub-optimal. With more expert demonstrations, we are able to improve the reward learning to some extent.
- RLHF is not able to decrease the number of demonstrations required by IRL when IRL is trained using sub-optimal demonstrations.
- **Limitations:** Experimenting on a simple model with only one type of algorithm for IRL and RLHF. Additionally, applying IRL and RLHF sequentially instead of blending them together.

References

[1] Stephen Adams, Tyler Cody, and Peter A. Beling.
A survey of inverse reinforcement learning.
Artificial Intelligence Review, 55(6):4307–4346, Feb 2022.

[2] Justin Fu, Katie Luo, and Sergey Levine.
Learning robust rewards with adversarial inverse reinforcement learning, 2018.

[3] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hullermeier.
A survey of reinforcement learning from human feedback, Dec 2023.

[4] Rafael Ris-Ala.
Fundamentals of Reinforcement Learning.
Springer International Publishing AG, 2023.

[5] Phil Winder.
Reinforcement learning.
2020.