INTERACTIVE INTELLIGENCE

EXPLORING THE SYNERGY BETWEEN INVERSE REINFORCEMENT LEARNING AND REINFORCEMENT LEARNING FROM HUMAN FEEDBACK FOR QUERY REDUCTION

Author: Ana Bătrîneanu

Supervisor: Antonio Mone
Responsible Professor: Luciano Cavalcante Siebert

TU Delft

1. INTRODUCTION

Reinforcement Learning is a powerful tool for problems that require sequential-decision-making. However, it often faces challenges due to the extensive need for reward engineering. **Reinforcement Learning from Human Feedback (RLHF) and Inverse Reinforcement Learning (IRL)** hold the promise of learning a reward function without manual encoding. While RLHF uses feedback to estimate a reward function, IRL learns from demonstrations, examples provided by a teacher.

IRL

- learns faster (given correct and diverse demonstrations)
- optimal demonstrations are hard to obtain

RLHF

- interactive feedback is generally believed to be easier to provide
- suffers from the curse of dimensionality
- learner has random behavior at early learning trials
- requires a large number of queries
- slow convergence to desired policy

We propose a learning framework (a simplified version inspired by [1]) in which these two approaches would potentially benefit from one another.

2. BACKGROUND

R = reward function
R(s, a) = immediate feedback received by performing action a in state s.
π = policy

Fig. 1: The basic RL setting

Proximal Policy Optimization [4]

- scalable, data efficient, robust (successful on a variety of problems without hyperparameter tuning)

RLHF with preference comparisons [2]

1. Reward Learning

- **Generate queries:** Present a human labeler (oracle) with two trajectory segments (A trajectory segment, σ, is a sequence of states and actions taken by the agent over a period of time within an environment)
- Oracle chooses preferred trajectory
- **Train a reward function approximator** with the answers provided by the oracle (typically done by minimizing the cross-entropy loss between the predicted and actual preferences)

2. RL training

- Running a deep RL algorithm using the currently trained reward function approximator
- **Derive a final policy**

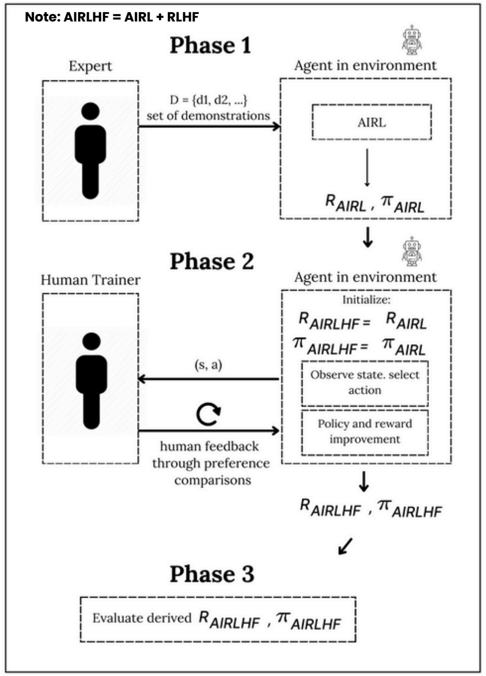
Adversarial IRL [3]

Adversarial Inverse Reinforcement Learning **(AIRL)** is a practical and scalable IRL algorithm based on an adversarial reward learning formulation. It uses a generative adversarial network (GAN) approach, where a discriminator (essentially acting as a reward model) and a generator (policy optimization agent) are trained simultaneously.

- recovers reward functions that are robust to changes in dynamics and generalize well
- performs well even in high-dimensional control tasks

(RQ) To what extent can IRL complement RLHF to reduce the number of queries RLHF needs?

3. METHODOLOGY



We run our experiments in the following way:

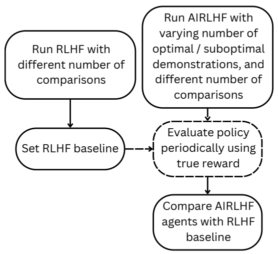


Fig. 2: Experimental setup

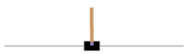


Fig. 3: CartPole-v0 environment

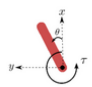
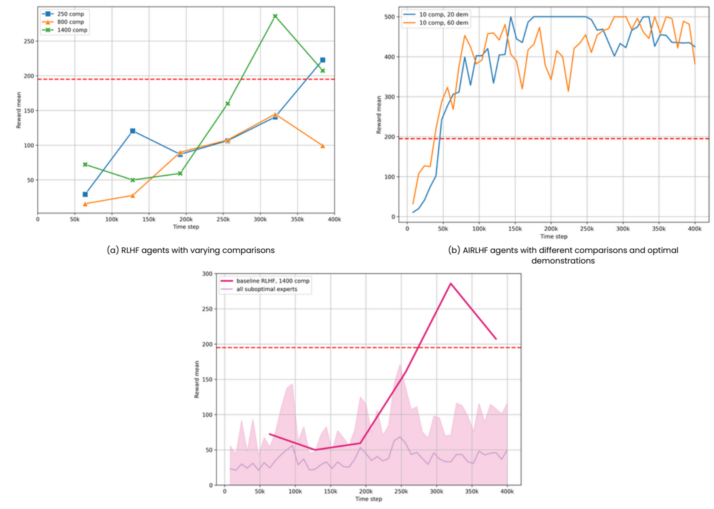


Fig. 4: Pendulum-v1 environment

4. RESULTS

CartPole-v0



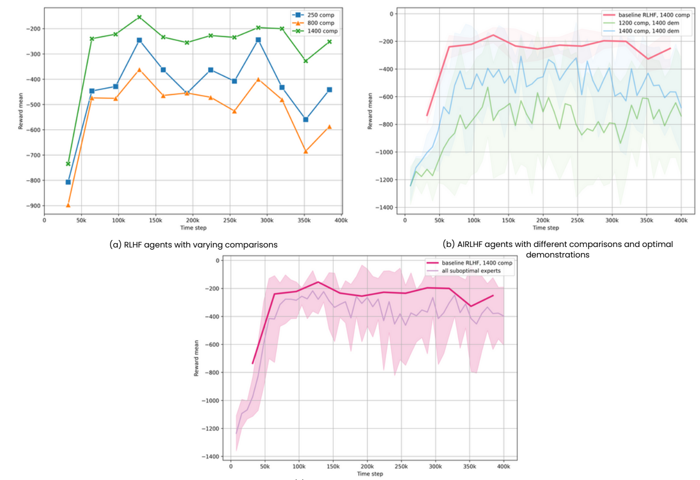
(a) RLHF agents with varying comparisons

(b) AIRLHF agents with different comparisons and optimal demonstrations

(c) A comparison between our selected RLHF baseline agent and the mean-curve of all suboptimal AIRLHF agents

Fig. 5: Results on CartPole-v0

Pendulum-v1



(a) RLHF agents with varying comparisons

(b) AIRLHF agents with different comparisons and optimal demonstrations

(c) A comparison between our selected RLHF baseline agent and the mean-curve of all suboptimal AIRLHF agents

Fig. 6: Results on Pendulum-v1

5. ANALYSIS & CONCLUSION

	AIRL optimal demonstrations + RLHF	AIRL suboptimal demonstrations + RLHF
CartPole-v0	≈ 93% decrease	no decrease
Pendulum-v1	no decrease	no decrease

	AIRL optimal demonstrations + RLHF	AIRL suboptimal demonstrations + RLHF
CartPole-v0	out-performing	worse / barely similar
Pendulum-v1	worse / barely similar	very similar

Suboptimal demonstrations: a large number of demonstrations is needed, equal or exceeding the number of comparisons used by the baseline RLHF agent, in order to be able to decrease the number of queries used by the AIRLHF agents.

Optimal demonstrations: the environment's dynamics and specifics greatly influence the outcome of our framework; in some environments, a larger number of optimal demonstrations and execution time is needed in order to provide a good initialization to RLHF; in other environments, a relatively small number of optimal demonstrations is enough to provide a good starting point to RLHF.

REFERENCES

[1] Ezzeddine, A.; Mourad, N.; Araabi, B. N.; and Ahmadabadi, M. N. 2018. Combination of learning from non-optimal demonstrations and feedbacks using inverse reinforcement learning and Bayesian policy improvement.
[2] Kaufmann, T.; Weng, P.; Bengs, V.; and H¨ullermeier, E. 2023. A Survey of Reinforcement Learning from Human Feedback
[3] Fu, J.; Luo, K.; and Levine, S. 2018. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning.
[4] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms.

A.Batrineanu@student.tudelft.nl