Conflict in the World of Inverse Reinforcement Learning: Investigating Inverse Reinforcement Learning with Conflicting Demonstrations

Petar Koev [1] Luciano Cavalcante Siebert [1] Antonio Mone [1]

[1]EEMCS, Delft University of Technology, The Netherlands

Introduction

- Inverse Reinforcement Learning (IRL) algorithms are closely related to Reinforcement Learning (RL) but instead, try to model the reward function from a given set of expert demonstrations.
- Most algorithms for IRL assume consistent demonstrations.
- Consistency is the assumption that all demonstrations follow the same underlying reward function and near-optimal policy.
- This, however, is not always the case. This study investigates the effect of conflicting demonstrations on IRL algorithms.

Research Questions

- ***To what extent can IRL learn rewards from conflicting demonstrations***
- *How does the degree of conflict between demonstrations affect IRL's ability to learn the reward function?*
- *Does the ratio of conflicting demonstrations influence IRL's ability to learn the reward?*
- *Does the complexity of the task influence IRL's ability to handle conflicting demonstrations?*
- *How do malicious expert demonstrations affect IRL?*

Definitions

Conflict

$$R_1(s, a, s') \neq R_2(s, a, s') \quad (1)$$

Malice

$$R_{\text{mal}}(s, a, s') = -R(s, a, s') \quad (2)$$

Methodology

Train RL Agents with Different Reward Functions Resulting in Conflicting and Malicious Policies

Generation of Trajectories

Train the AIRL Algorithm [1]

Policy Evaluation

Results Conflict

Comparison of agents trained with different ratios of conflicting demonstrations. All agents achieve comparable results.

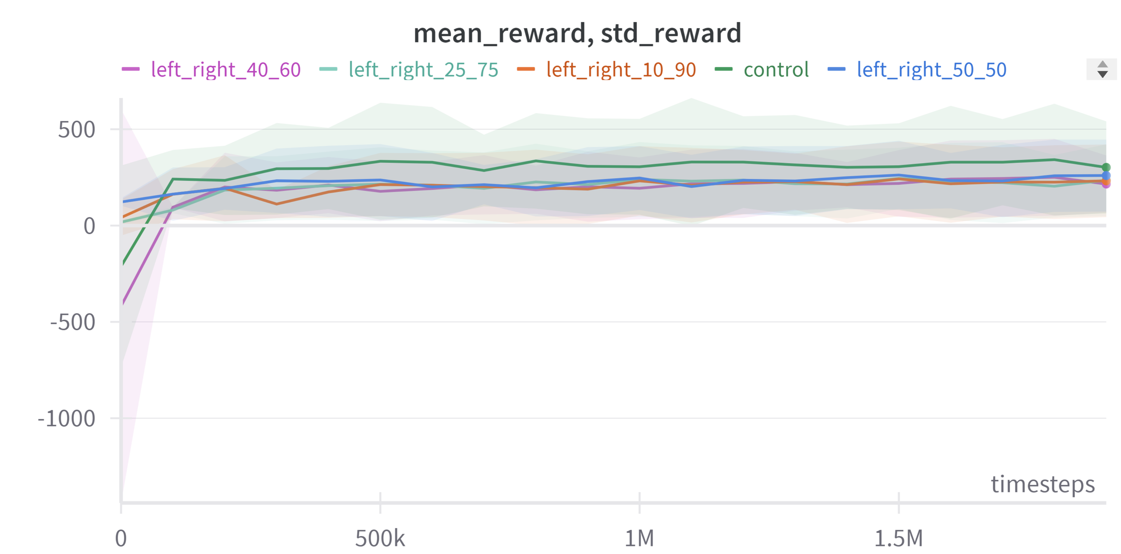

Figure 2:Graph showcasing the learning of AIRL agents in the LunarLander-v2 environment.

Run	Final Reward	Final Std
control	263.9	57.3
left_right_50_50	274.3	52.2
left_right_40_60	179.8	112.6
left_right_25_75	214.5	94.9
left_right_10_90	225.4	88.2

Table 1:Comparison of final mean reward and final mean standard deviation for the LunarLander-v2 environment.

Results Malice

Figure 1 shows that the agent with a 10% split of malicious and expert demonstrations achieves the same results as the control agent, while the other two agents fail to learn the reward function.

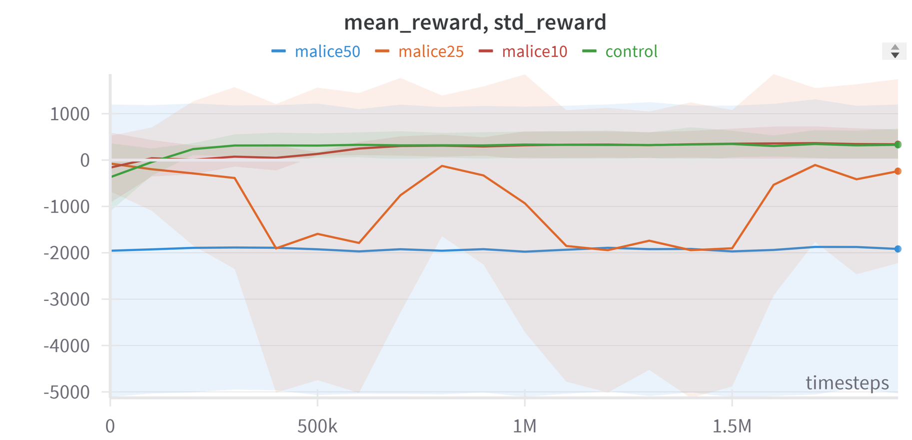

Figure 1:Comparison of agents trained with different ratios of malicious demonstrations

Unexpected Results

Our observations are that AIRL averages out the two conflicting reward functions as shown by the engine usage of the mo-lunar-lander-v2 environment in Table 2

# Main Engine Use	# Side Engines Use	Run Name
70	21	control
100	14	main_side_90_10
75	48	main_side_75_25
86	54	main_side_50_50
70	38	main_side_25_75

Table 2:Engine usage statistics for different runs.

However, when we trained agents in the resource-gathering-v0 environment, agents preferred only one of the objectives and went only for it as shown in Table 3.

Run	Final Reward	Final Std
control	1.8	0.8
gem_gold_50_50	1.0	0.0
gem_gold_40_60	1.0	0.0
gem_gold_25_75	1.0	0.0
gem_gold_10_90	0.8	0.5

Table 3:Comparison of final mean reward and final mean standard deviation for the resource-gathering-v0 environment.

This is explained by the discriminator behaviour shown in Figure 3.

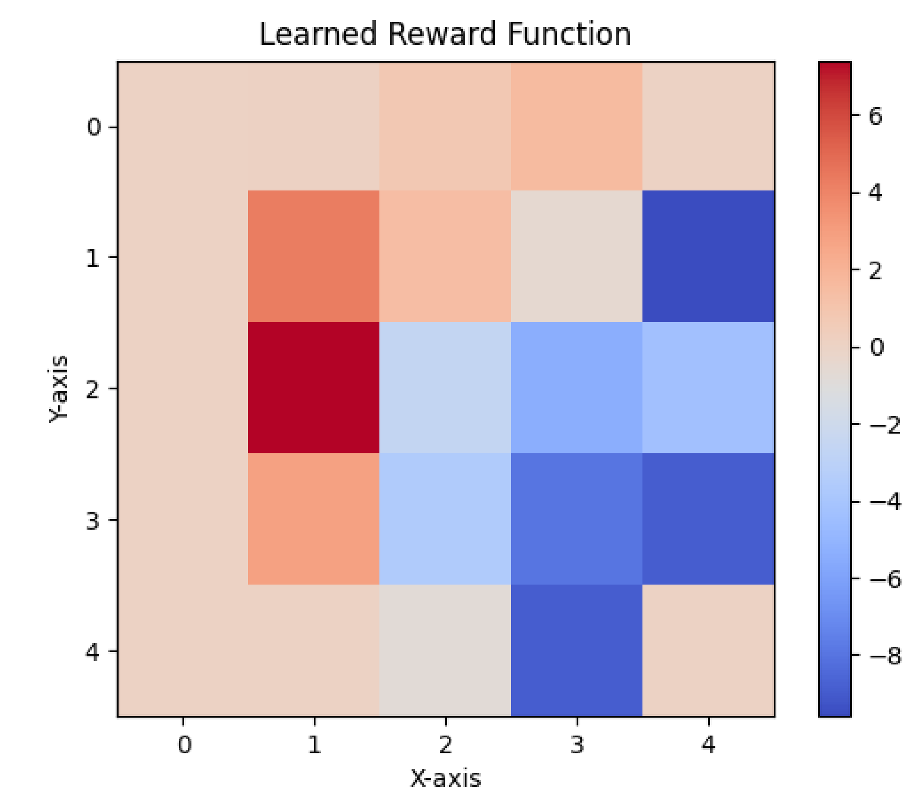

Figure 3:Plot of the rewards predicted by the reward net of the gem_gold_50_50 agent.

Conclusion

- IRL algorithms can learn optimal policies even with conflicting demonstrations.
- As the degree of conflict intensifies, it becomes more challenging for the algorithm to learn.
- Malicious demonstrations had a great impact on performance even when they constituted only a small portion of the demonstrations.

References

1. J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," CoRR, vol. abs/1710.11248, 2017. [Online]. Available: http://arxiv.org/abs/1710.11248

TU Delft Delft University of Technology