역강화학습을 위한 전이가능 보상함수의 분해

Transferable-Reward Decomposition for Inverse Reinforcement Learning

장재휘[1]·박대형[2†]

Jaehwi Jang[1], Daehyung Park[2†]

Abstract Reward decomposition is the process of identifying reward types for explaining the decisions of reinforcement learning. However, the decomposition is nontrivial in that a total reward can be formulated in a large number of possible sub-rewards (i.e., ill-posed problem). In this work, we introduce a transferable reward-decomposition method for inverse reinforcement learning (IRL) that returns a unique pair of primary goal and its residual rewards, separately. Our method lowers the computational complexity of reward decomposition by reusing the exploration results of IRL. Through evaluations in two simulated environments, our method successfully decomposed residual reward component and transferable to novel environments.

Keywords: Inverse Reinforcement Learning, Reward Decomposition

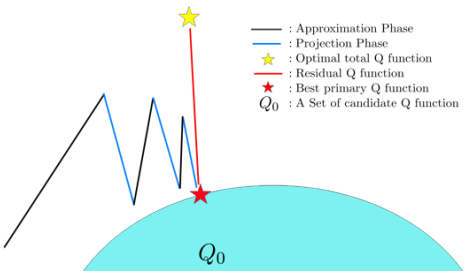
※ This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT)
(No.2021R1C1C1004368 and 2021R1A4A3032834)
1. Graduate student, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea (wognl0402@kaist.ac.kr)
2. Assistant Professor, School of Computing, KAIST, Daejeon, Korea (daehyung@kaist.ac.kr)

1. Introduction

Inverse Reinforcement Learning (IRL) is a problem of identifying a reward function from demonstrations. However, most identification process does not expose the underlying structure of the reward value, though the structure information can provide better understanding of an assigned task and its transferability. For instance, suppose that a safe-driving task. A recovered reward function may consist of goal-reaching reward and obstacle-crashing penalty functions. By replacing the reaching reward with a new one, we will be able to easily transfer the safety behavior to the new setup without conflicts.

The decomposition of a reward function is often ambiguous and costly in that there can be infinitely large number of subreward combinations (i.e., ill-posed problem). Further, conventional value-wise comparison of meaningful subreward candidates requires computationally expensive optimization doubling the recovery complexity of IRL [1]. A desired method should show the ability to find meaningful sub-reward functions without prior knowledge in an efficient manner.

In this work, we present a novel reward-decomposition method that finds explainable and transferable sub-reward functions. Assuming the form of a primary reward function is given, our method finds the residual reward function maximally disentangling from the primary one. Our method requires lower computation cost by reusing the state-action distribution outputs from IRL. Our method can work on top of any existing IRL frameworks. We evaluate our method with a state-of-the-art baseline method in two simulated environments and show our method outperforms.



[Fig. 1] Approximation phase and Reward Projection phase guides to optimal primary Q function, Q_p^*.

2. Method

We first define the problem of finding primary and residual reward pair (r_p, r_r) with a Markov decision process (MDP). Our MDP is a tuple (S, A, P, R, γ) with an unknown reward function $R = r_p + r_r$, where the symbols S, A, P, and γ denote the state space, the action space, a stochastic transition function and a discount

factor, respectively. Our proposed method finds R that equals that of expert demonstrations maximally disentangling the sub-rewards r_p and r_r. Note that we assume a set of demonstrations D_E and a set of reward candidates R_p for primary reward are given (i.e., $r_p \in R_p$). We find the residual reward r_r by subtracting the primary reward r_p from the total reward R (i.e., $r_r = R - r_p$). We can estimate R and r_p given the demonstrations D_E via IRL. To maximally disentangling r_p from r_r, we introduce two phases of r_p estimation method:

1) Approximation: This phase learns a single Q function with implicitly learned reward r_p using Inverse soft-Q learning[2] (IQ-Learn), which helps not only providing additional optimization projective that we want but also sharing the exploration experience with another IRL process. Our method particularly finds a Q function that maximizes the disentanglement while regulating non-zero rewards of r_p and r_r given the demonstrations. We denote μ^{π} as the state-action distribution of a policy π. π_E refers to the expert policy while π_p and π_r denote optimal policies given r_p and r_r, respectively. We define the objective-loss function L as

$$L = D_f(\mu^{\pi_E} || \mu^{\pi_0}) \geq \frac{1}{2}[D_f(\mu^{\pi_E} || \mu^{\pi_0}) + D_f(\mu^{\pi_E} || \mu^{\pi'}) - D_f(|\mu^{\pi'})],$$

where D_f is a measure of symmetric statistical divergence. In this work, we use Jensen-Shannon divergence. By plugging the function L as an objective function of IQ-Learn, we obtain a Q_p with the primary reward r_p.

2) Projection: We minimize the difference between the approximated Q_p and the optimal Q_p^* by projecting Q_p from the approximation phase onto the space of Q functions from the primary reward candidates R_p. To do that, we update Q_p minimizing a regulated soft Bellman error:

$$E\left(Q_p, r_p(s,a,s')\right) = \left(Q_p(s,a) - \gamma V_p(s') - r_0(s,a,s')\right)^2$$

where V_p is a state-value function. We compute an error per sample in demonstrations as well as explorations. We then update Q_p and r_p using their gradients. Finally, we obtain the target residual reward r_r $(= R - r_p)$ from the best r_p and R computed from another IQ-Learn.

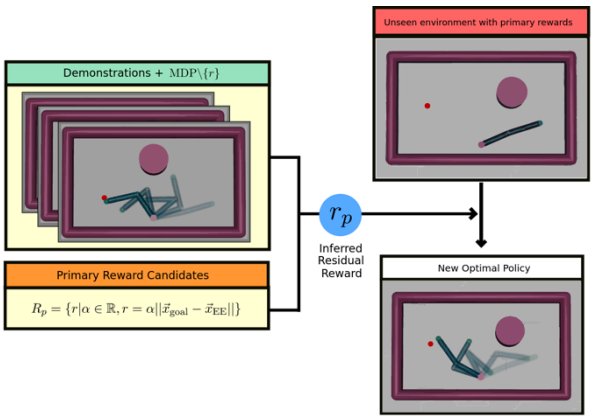
3. Experiments

We tested our method on a gridworld and a synthetic simulated robotics task. Our objective is to answer the following questions: 1) Can our method decompose into disentangled sub-rewards? 2) Does our method learn transferrable rewards?

Gridworld environment: In the gridworld environment, there are three goals with a reward. By setting one of the signal as primary reward, we can qualitatively verify that two additional goal signals are inferred with a negligible primary goal signal.

Mujoco simulation: In the synthetic robotic reaching tasks, the agent must bring the end-effector to the desired position while avoid the collision with the obstacles. The agent has given a set of candidates for primary reward (distance between end-effector and the goal). After inferring a residual reward, we transfer the residual reward to other similar MDPs but with different settings to verify transferability. First, we choose the same MDP to check a consistency of the residual reward. Second, we choose the MDP with different goal and obstacle configuration. After finding new optimal policy for a new MDP, we check the success case (reaching the goal without hitting the obstacles) of baseline method and our methods.

[Fig. 2] Experiment in gridworld. All rewards are normalized to unity. From the left, ground truth, total reward from IRL, (primary and residual reward from our method.



[Fig. 3] Simulated mujoco robotics tasks. To check the transferability, the residual rewards are used in novel MDP.

[Table 1] Testing Consistency and Transferability Testing of Residual Reward in a synthetic robotics tasks.

Rate of Success (%)	The same MDP with demonstrations	Randomized MDP
Our method	50.9	43.8
IRL with regression	41.3	7.8

4. Conclusion

We introduced a novel method for transferable-reward decomposition for IRL. We successfully decomposed a residual reward and a primary reward from given demonstrations.

References

[1] Christopher Grimm, Satinder Singh, “Learning Independently-Obtainable Reward Functions” *ArXiv preprint*, arXiv:1901.08649, 2019.

[2] Garg, Divyansh and Chakraborty, Shuvam and Cundy, Chris and Song, Jiaming and Ermon, Stefano, “IQ-Learn: Inverse soft-Q Learning for Imitation,” In *Conf. on Neural Information Processing System,* 2021.