Generative Adversarial Imitation Learning from Failed Experiences* (Student Abstract)

Jiacheng Zhu,[1] Jiahao Lin,[1] Meng Wang,[2]
Yingfeng Chen,[2] Changjie Fan,[2] Chong Jiang,[1] Zongzhang Zhang[3]
[1]School of Computer Science and Technology, Soochow University, Suzhou 215006, China
[2]Fuxi AI Lab, Netease, Hangzhou 310052, China
[3]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
jczhu0@stu.suda.edu.cn, {wzljh3148, jch0ng}@outlook.com,
{wangmeng02, chenyingfeng1, fanchangjie}@corp.netease.com, zzzhang@nju.edu.cn

Abstract

Imitation learning provides a family of promising methods that learn policies from expert demonstrations directly. As a model-free and on-line imitation learning method, generative adversarial imitation learning (GAIL) generalizes well to unseen situations and can handle complex problems. In this paper, we propose a novel variant of GAIL called GAIL from failed experiences (GAILFE). GAILFE allows an agent to utilize failed experiences in the training process. Moreover, a constrained optimization objective is formalized in GAILFE to balance learning from given demonstrations and from self-generated failed experiences. Empirically, compared with GAIL, GAILFE can improve sample efficiency and learning speed over different tasks.

Introduction

Imitation learning provides a promising way for an agent to learn a decision model by imitating the expert demonstrations, and has achieved remarkable successes in a wide range of problems. Generative adversarial imitation learning (GAIL) (Ho and Ermon 2016) is a state-of-the-art imitation learning method, which is able to solve complex and high-dimensional problems. However, it needs more expert demonstrations as the environment becomes more complecated. But in some areas it is difficult to get perfect expert demonstrations. In the process of collecting expert demonstrations, there are many failed experiences which are discarded in the end. During the training process, the agent also generates lots of failed experiences. However, GAIL is not able to make good use of these failed experiences. Besides, GAIL requires a significant number of interactions with the environment to achieve promising learning performance because of the nature of model-free and on-line learning, which makes it even harder to be applied in practice. Inverse reinforcement learning (IRL) from failure (Shiarlis, Messias, and Whiteson 2016) learns a policy using both successful and failed demonstrations. However, IRL algorithms require reinforcement learning in an inner loop, which makes it extremely expensive to run.

In this paper, with the aim of closing this gap and scaling well to real-world problems, we propose a novel framework on top of GAIL, i.e., GAIL from failed experiences (GAILFE). Unlike GAIL, GAILFE takes advantage of failed experiences. We store failed experiences that should be avoided by the agent in a replay buffer, and replay the failed experiences in the training process. In practice, there is a challenge in balancing learning from the expert demonstrations and the failed experiences. In our method, we formalize a constrained optimization objective to solve it.

Method

Consider how an agent can learn a good policy with only a set of expert demonstrations τ_{E}. Here, τ_{E} is a set of trajectories, each of which consists of a sequence of state-action pairs. In GAIL, an agent mimics the behavior of expert by matching the distribution of generated state-action pairs $\rho_{\pi_\theta}(s,a)$ with expert's distribution $\rho_{\pi_E}(s,a)$. The formal objective of GAIL can be denoted as:

$$\min_{\theta}\max_{\omega} J_{\texttt{GAIL}} = \mathbb{E}_{(s,a)\sim\pi_\theta}[\log(D_\omega(s,a))] + \mathbb{E}_{(s,a)\sim\pi_E}[\log(1-D_\omega(s,a))] + \lambda_H H(\pi_\theta) \tag{1}$$

where D_ω denotes the binary discriminator, parameterized by ω. The discriminator tries to distinguish the expert state-action pairs from the ones generated by policy π_θ. $H(\pi_\theta) \triangleq \mathbb{E}_{\pi_\theta}[-\log\pi_\theta(a|s))]$ denotes the discounted causal entropy of π_θ (Bloem and Bambos 2014) and λ_H is the coefficient on it. The policy parameterized by θ plays the role as a generator which generates samples to confuse discriminator. The optimization over GAIL objective is performed by alternating between increasing $J_{\texttt{GAIL}}$ with respect to the discriminator parameters ω, and conducting a trust region policy optimization (TRPO) (Schulman et al. 2015) step to decrease $J_{\texttt{GAIL}}$ with respect to the policy parameters θ using the reward function $-\log(D_\omega(s,a))$.

The framework of GAILFE is a little different from that of GAIL in that it adds a replay buffer β_{F} to store failed experiences. In each iteration, we randomly sample a batch of samples from the replay buffer for updating the discriminator. We assume the access to an annotator who processes the

*Corresponding author: Zongzhang Zhang. This work is in part supported by the Natural Science Foundation of China (61876119) and the Natural Science Foundation of Jiangsu (BK20181432).

prior knowledge of which trajectory is failure. We believe that such an assumption can be easily satisfied, because the prior knowledge of failure is usually common sense. In practice, our method only stores last N failed trajectories in the replay buffer.

By using failed experiences to train an discriminator, it can make the discriminator more sensitive to failed behaviors. Then the discriminator gives rewards as little as possible to the agent when it generates such failed state-action pairs. The process of training a discriminator with failed experiences can be formulated as:

$$\mathbb{E}_{(s,a)\sim\beta_F}[\log(D_\omega(s,a))] > Z_F \tag{2}$$

where Z_F is a variable that adjusts the intensity of punishment upon failed experiences. Actually, the policy update in GAILFE is consistent with GAIL. Thus, the reward function for updating a policy is also $-\log(D_\omega(s,a))$. So the greater Z_F is, the less rewards the agent receives when generating such failed state-action pairs. Combining it with GAIL, we get a constrained objective of GAILFE:

$$\min_\theta \max_\omega J_{\texttt{GAIL}} \quad s.t.\ \mathbb{E}_{(s,a)\sim\beta_F}[\log(D_\omega(s,a))] > Z_F \tag{3}$$

The optimization problem of GAILFE can be solved by further transforming it to the Lagrangian duality:

$$\begin{aligned} & \min_\theta \max_\omega L(\pi_\theta, D_\omega, \lambda_F) \\ = \ & J_{\texttt{GAIL}} + \lambda_F\{\mathbb{E}_{(s,a)\sim\beta_F}[\log(D_\omega(s,a))] - Z_F\} \end{aligned} \tag{4}$$

where λ_F denotes the Lagrangian multiplier and it plays a key role in balancing learning from both expert demonstrations and failed experiences.

In our algorithm, the update of GAILFE alternates between increasing $L(\pi_\theta, D_\omega, \lambda_F)$ with respect to ω, and decreasing $L(\pi_\theta, D_\omega, \lambda_F)$ with respect to θ.

Experiments

In this section, we compare the performance of GAILFE with GAIL on two physics-based control tasks simulated with MuJoCo, HalfCheetah and Hopper. Each task comes with a reward function defined in the OpenAI Gym. For these tasks, we generate expert demonstrations by running the proximal policy optimization (PPO) algorithm (Schulman et al. 2017) on these reward functions. When we get expert demonstrations, there will include some failed experiences. We add the failed experiences to the replay buffer. In these experimental environments, we use 30 expert trajectories for both of them, and each trajectory contains 1000 state-action pairs. GAIL trained with expert demonstrations only, and GAILFE trained with both expert demonstrations and failed experiences. The neural networks used for representing a policy and a discriminator are built with 2 hidden layers, where there are 100 hidden units for each hidden layer with tanh activation. In order to satisfy the demand of significance test, we set up 3 different seeds for each task. The hyper-parameters N, Z_F are set as 1500 and $\log(1/2)$, respectively.

The performance of our method is examined based on the learning curves presented in Figure 1. It is clear that our method converges faster.

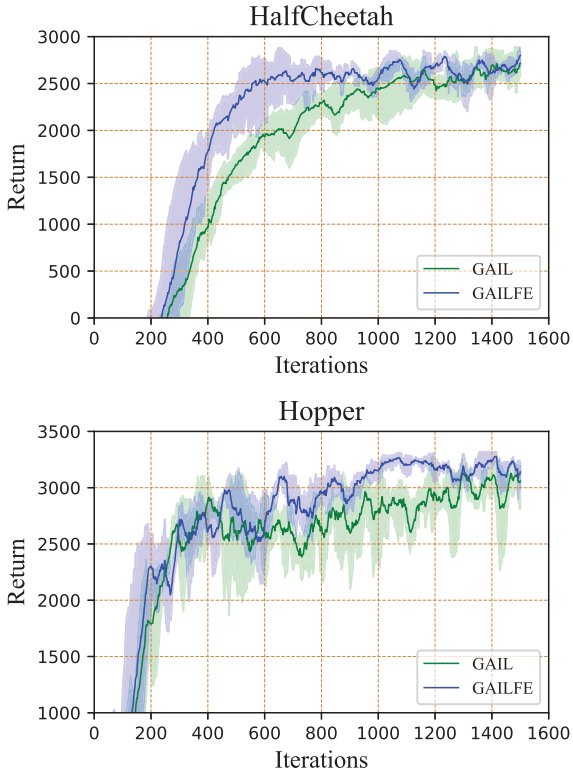

Figure 1: Learning curve of HalfCheetah and Hopper. Each iteration consists of 1024 time steps.

Conclusion and Future Work

In this paper, we propose a novel algorithm called GAILFE, which can improve sample efficiency and learning speed. We use failed experiences generated by an agent in the training process for training a sensitive discriminator to assign less rewards to the failed behavior. In this way, the agent can avoid repeatedly exploring some failed behaviors. As a future work, we consider adding successful samples that an agent generated during the training process to expert demonstrations to further improve sample efficiency.

References

Bloem, M., and Bambos, N. 2014. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *CDC*, 4911–4916.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *NIPS*, 4565–4573.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *ICML*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*.

Shiarlis, K.; Messias, J. V.; and Whiteson, S. 2016. Inverse reinforcement learning from failure. In *AAMAS*, 1060–1068.