On Computation and Generalization of Generative Adversarial Imitation Learning

Minshuo Chen[1], **Yizhou Wang**[4], **Tianyi Liu**[1], **Zhuoran Yang**[2], **Xingguo Li**[2], **Zhaoran Wang**[3] **and Tuo Zhao**[1]
Georgia Tech[1], Princeton University[2], Northwestern University[3], Xi'an Jiaotong University[4]

Abstract

Generative Adversarial Imitation Learning (GAIL) is a powerful and practical approach for learning sequential decision-making policies. Different from Reinforcement Learning (RL), GAIL takes advantage of demonstration data by experts (e.g., human), and learns both the policy and reward function of the unknown environment. Despite the significant empirical progresses, the theory behind GAIL is still largely unknown. The major difficulty comes from the underlying temporal dependency of the demonstration data and the minimax computational formulation of GAIL without convex-concave structure. To bridge such a gap between theory and practice, this paper investigates the theoretical properties of GAIL. Specifically, we show: (1) For GAIL with general reward parameterization, the generalization can be guaranteed as long as the class of the reward functions is properly controlled; (2) For GAIL, where the reward is parameterized as a reproducing kernel function, GAIL can be efficiently solved by stochastic first order optimization algorithms, which attain sublinear convergence to a stationary solution. To the best of our knowledge, these are the first results on statistical and computational guarantees of imitation learning with reward/policy function approximation.

1 Introduction

As various robots (Tail et al., 2018), self-driving cars (Kuefler et al., 2017), unmanned aerial vehicles (Pfeiffer et al., 2018) and other intelligent agents are applied to complex and unstructured environments, programming their behaviors/policy has become increasingly challenging. These intelligent agents need to accommodate a huge number of tasks with unique environmental demands. To address these challenges, many reinforcement learning (RL) methods have been proposed for learning sequential decision-making policies (Sutton et al., 1998; Kaelbling et al., 1996; Mnih et al., 2015). These RL methods, however, heavily rely on human expert domain knowledge to design proper reward functions. For complex tasks, which are often difficult to describe formally, these RL methods become impractical.

The Imitation Learning (IL, Argall et al. (2009); Abbeel and Ng (2004)) approach is a powerful and practical alternative to RL. Rather than having a human expert handcrafting a reward function for learning the desired policy, the imitation learning approach only requires the human expert to demonstrate the desired policy, and then the intelligent agent (a.k.a. learner) learns to match the demonstration. Most of existing imitation learning methods fall in two categories: 1). Behavioral Cloning (BC, Pomerleau (1991)) and 2). Inverse Reinforcement Learning (IRL, Russell (1998); Ng et al. (2000); Finn et al. (2016); Levine and Koltun (2012)). However, BC often suffers from poor generalization, and IRL often fails to scale to large and high dimensional environments.

More recently, Ho and Ermon (2016) propose a Generative Adversarial Imitation Learning (GAIL) method, which obtains significant performance gains over existing IL methods in imitating complex expert policies in large and high-dimensional environments. GAIL generalizes IRL by formulating

Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019, Vancouver, Canada.

the IL problem as minimax optimization, which can be solved by alternating gradient-type algorithms in a more scalable and efficient manner.

Specifically, we consider an infinite horizon Markov Decision Process (MDP), where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, P denotes the Markov transition kernel, r^* denotes the reward function, and p_0 denotes the distribution of the initial state. We assume that the Markov transition kernel P is fixed and there is an unknown expert policy $\pi^*: \mathcal{S} \to \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ denotes the set of distributions over the action space. As can be seen, $\{s_t\}_{t=0}^{T-1}$ essentially forms a Markov chain with the transition kernel induced by π^* as $P^{\pi^*}(s, s') = \sum_{a\in\mathcal{A}} \pi^*(a \,|\, s) \cdot P(s' \,|\, s, a)$. Given n demonstration trajectories from π^* denoted by $\{s_t^{(i)}, a_t^{(i)}\}_{t=0}^{T-1}$, where $i = 1, ..., n$, $s_0 \sim p_0$, $a_t \sim \pi^*(\cdot \,|\, s_t)$, and $s_{t+1} \sim P(\cdot \,|\, s_t, a_t)$, GAIL aims to learn π^* by solving the following minimax optimization problem,

$$\min_\pi \max_{r\in\mathcal{R}} [\mathbb{E}_\pi r(s, a) - \mathbb{E}_{\pi_n^*} r(s, a)], \tag{1}$$

where $\mathbb{E}_\pi[r(s, a)] = \lim_{T\to\infty} \mathbb{E}[\frac{1}{T}\sum_{t=0}^{T-1} r(s_t, a_t)|\pi]$ denotes the average reward under the policy π when the reward function is r, and $\mathbb{E}_{\pi_n^*}[r(s, a)] = \frac{1}{nT}\sum_{i=1}^{n}\sum_{t=0}^{T-1}[r(s_t^{(i)}, a_t^{(i)})]$ denotes the empirical average reward over the demonstration trajectories. As shown in (1), GAIL aims to find a policy, which attains an average reward similar to that of the expert policy with respect to any reward belonging to the function class $\mathcal{R}$.

For large and high-dimensional imitation learning problems, we often encounter infinitely many states. To ease computation, we need to consider function approximations. Specifically, suppose that for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$, there are feature vectors $\psi_s \in \mathbb{R}^{d_\mathcal{S}}$ and $\psi_a \in \mathbb{R}^{d_\mathcal{A}}$ associated with a and s, respectively. Then we can approximate the policy and reward as $\pi(\cdot|s) = \widetilde{\pi}_\omega(\psi_s)$ and $r(s, a) = \widetilde{r}_\theta(\psi_s, \psi_a)$, where $\widetilde{\pi}$ and $\widetilde{r}$ belong to certain function classes (e.g. reproducing kernel Hilbert space or deep neural networks, Ormoneit and Sen (2002); LeCun et al. (2015)) associated with parameters ω and θ, respectively. Accordingly, we can optimize (1) with respect to the parameters ω and θ by scalable alternating gradient-type algorithms.

Although GAIL has achieved significant empirical progresses, its theoretical properties are still largely unknown. There are three major difficulties when analyzing GAIL: 1). There exists temporal dependency in the demonstration trajectories/data due to their sequential nature (Howard, 1960; Puterman, 2014; Abounadi et al., 2001); 2). GAIL is formulated as a minimax optimization problem. Most of existing learning theories, however, focus on empirical risk minimization problems, and therefore are not readily applicable (Vapnik, 2013; Mohri et al., 2018; Anthony and Bartlett, 2009); 3). The minimax optimization problem in (1) does not have a convex-concave structure, and therefore existing theories in convex optimization literature cannot be applied for analyzing the alternating stochastic gradient-type algorithms (Willem, 1997; Ben-Tal and Nemirovski, 1998; Murray and Overton, 1980; Chambolle and Pock, 2011; Chen et al., 2014).

To bridge such a gap between practice and theory, we establish the generalization properties of GAIL and the convergence properties of the alternating mini-batch stochastic gradient algorithm for solving (1). Specifically, our contributions can be summarized as follows:

• We formally define the generalization of GAIL under the "so-called" $\mathcal{R}$-reward distance, and then show that the generalization of GAIL can be guaranteed under reward distance as long as the class of the reward functions is properly controlled;

• We provide sufficient conditions, under which an alternating mini-batch stochastic gradient algorithm can efficiently solve the minimax optimization in (1), and attains sublinear convergence to a stationary solution.

Our work is related to Syed et al. (2008); Cai et al. (2019). Syed et al. (2008) study the generalization and computational properties of apprenticeship learning. Since they assume that the state space of the underlying Markov decision process is finite, they do not consider any reward/policy function approximations; Cai et al. (2019) study the computational properties of imitation learning under a simple control setting. Their assumption on linear policy and quadratic reward is very restrictive, and does not hold for many real applications.

2 Generalization of GAIL

To analyze the generalization properties of GAIL, we first assume that we can access an infinite number of the expert's demonstration trajectories (underlying population), and that the reward

function is chosen optimally within some large class of functions. This allows us to remove the maximum operation from (1), which leads to an interpretation of how and in what sense the resulting policy is close to the true expert policy. Before we proceed, we first introduce some preliminaries.

Definition 1 (Stationary Distribution)**.** Note that any policy π induces a Markov chain on $\mathcal{S} \times \mathcal{A}$. The transition kernel is given by

$$P_\pi(s', a' \,|\, s, a) = \pi(a' \,|\, s') \cdot P(s' \,|\, s, a), \quad \forall (s, a), (s', a') \in \mathcal{S} \times \mathcal{A}.$$

When such a Markov chain is aperiodic and recurrent, we denote its stationary distribution as ρ_π.

Note that a policy π is uniquely determined by its stationary distribution ρ_π in the sense that

$$\pi(a \,|\, s) = \rho_\pi(s, a) / \textstyle\sum_{a \in \mathcal{A}} \rho_\pi(s, a).$$

Then we can write the expected average reward of $r(s, a)$ under the policy π as

$$\mathbb{E}_\pi[r(s,a)] = \lim_{T\to\infty} \mathbb{E}\big[\tfrac{1}{T}\textstyle\sum_{t=0}^{T-1} r(s_t, a_t) \big| \pi\big] = \mathbb{E}_{\rho_\pi}\big[r(s,a)\big] = \textstyle\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \rho_\pi(s,a) \cdot r(s,a).$$

We further define the $\mathcal{R}$-distance between two policies π and π' as follows.

Definition 2. Let $\mathcal{R}$ denote a class of symmetric reward functions from $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}$, i.e., if $r \in \mathcal{R}$, then $-r \in \mathcal{R}$. Given two policy π' and π, the $\mathcal{R}$-distance for GAIL is defined as

$$d_{\mathcal{R}}(\pi, \pi') = \sup_{r \in \mathcal{R}} [\mathbb{E}_\pi r(s,a) - \mathbb{E}_{\pi'} r(s,a)].$$

The $\mathcal{R}$-distance over policies for Markov decision processes is essentially an Integral Probability Metric (IPM) over stationary distributions (Müller, 1997). For different choices of $\mathcal{R}$, we have various $\mathcal{R}$-distances. For example, we can choose $\mathcal{R}$ as the class of all 1-Lipschitz continuous functions, which yields that $d_{\mathcal{R}}(\pi, \pi')$ is the Wasserstein distance between ρ_π and $\rho_{\pi'}$ (Vallender, 1974). For computational convenience, GAIL and its variants usually choose $\mathcal{R}$ as a class of functions from some reproducing kernel Hilbert space, or a class of neural network functions.

Definition 3. Given n demonstration trajectories from time 0 to $T-1$ obtained by an expert policy π^* denoted by $(s_t^{(i)}, a_t^{(i)})_{t=0}^{T-1}$, where $i = 1, ..., n$, a policy $\widehat{\pi}$ learned by GAIL generalizes under the $\mathcal{R}$-distance $d_{\mathcal{R}}(\cdot, \cdot)$ with generalization error ϵ, if with high probability, we have

$$|d_{\mathcal{R}}(\pi_n^*, \widehat{\pi}) - d_{\mathcal{R}}(\pi^*, \widehat{\pi})| \le \epsilon,$$

where $d_{\mathcal{R}}(\pi_n^*, \widehat{\pi})$ is the empirical $\mathcal{R}$-distance between π^* and $\widehat{\pi}$ defined as

$$d_{\mathcal{R}}(\pi_n^*, \widehat{\pi}) = \sup_{r\in\mathcal{R}} [\mathbb{E}_{\pi_n^*} r(s,a) - \mathbb{E}_{\widehat{\pi}} r(s,a)] \;\text{ with }\; \mathbb{E}_{\pi_n^*}[r(s,a)] = \tfrac{1}{nT}\textstyle\sum_{i=1}^{n}\sum_{t=0}^{T-1}[r(s_t^{(i)}, a_t^{(i)})].$$

The generalization of GAIL implies that the $\mathcal{R}$-distance between the expert policy π^* and the learned policy $\widehat{\pi}$ is close to the empirical $\mathcal{R}$-distance between them. Our analysis aims to prove the former distance to be small, whereas the latter one is what we attempts to minimize in practice.

We then introduce the assumptions on the underlying Markov decision process and expert policy.

Assumption 1. Under the expert policy π^*, $(s_t, a_t)_{t=0}^{T-1}$ forms a stationary and exponentially β-mixing Markov chain, i.e.,

$$\beta(k) = \sup_n \mathbb{E}_{B \in \sigma_0^n} \sup_{A \in \sigma_{n+k}^\infty} |\mathbb{P}(A|B) - \mathbb{P}(A)| \le \beta_0 \exp(-\beta_1 k^\alpha),$$

where β_0, β_1, α are positive constants, and σ_i^j is the σ-algebra generated by $(s_t, a_t)_{t=i}^j$ for $i \le j$.

Moreover, for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$, there are feature vectors $\psi_s \in \mathbb{R}^{d_\mathcal{S}}$ and $\psi_a \in \mathbb{R}^{d_\mathcal{A}}$ associated with a and s, respectively, and ψ_s and ψ_a are uniformly bounded, where

$$\|\psi_s\|_2 \le 1 \quad \text{and} \quad \|\psi_a\|_2 \le 1, \quad \forall s \in \mathcal{S} \quad \text{and} \quad \forall a \in \mathcal{A}.$$

Assumption 1 requires the underlying MDP to be ergodic (Levin and Peres, 2017), which is a commonly studied assumption in exiting reinforcement learning literature on maximizing the expected average reward (Strehl and Littman, 2005; Li et al., 2011; Brafman and Tennenholtz, 2002; Kearns and Singh, 2002). The feature vectors associated with a and s allow us to apply function approximations to parameterize the reward and policy functions. Accordingly, we write the reward function as $r(s, a) = \widetilde{r}(\psi_s, \psi_a)$, which is assumed to be bounded.

Assumption 2. The reward function class is uniformly bounded, i.e., $\|r\|_\infty \le B_r$ for any $r \in \mathcal{R}$.

Now we proceed with our main result on generalization properties of GAIL. We use $\mathcal{N}(\mathcal{R}, \epsilon, \|\cdot\|_\infty)$ to denote the covering number of the function class $\mathcal{R}$ under the ℓ_∞ distance $\|\cdot\|_\infty$.

Theorem 1 (Main Result)**.** Suppose Assumptions 1-2 hold, and the policy learned by GAIL satisfies

$$d_{\mathcal{R}}(\pi_n^*, \widehat{\pi}) - \inf_\pi d_{\mathcal{R}}(\pi_n^*, \pi) < \epsilon,$$

where the infimum is taken over all possible learned policies. Then with probability at least $1-\delta$ over the joint distribution of $\{(a_t^{(i)}, s_t^{(i)})_{t=0}^{T-1}\}_{i=1}^n$, taking $\zeta = (\beta_1^{-1} \log \frac{\beta_0 T}{\delta})^{\frac{1}{\alpha}}$, we have

$$d_{\mathcal{R}}(\pi^*, \widehat{\pi}) - \inf_\pi d_{\mathcal{R}}(\pi^*, \pi) \le O\left(\frac{B_r}{\sqrt{nT/\zeta}} \sqrt{\log \mathcal{N}\left(\mathcal{R}, \sqrt{\tfrac{\zeta}{nT}}, \|\cdot\|_\infty\right)} + B_r \sqrt{\frac{\log(1/\delta)}{nT/\zeta}} \right) + \epsilon.$$

Theorem 1 implies that the policy $\widehat{\pi}$ learned by GAIL generalizes as long as the complexity of the function class $\mathcal{R}$ is well controlled. To the best of our knowledge, this is the first result on the generalization of imitation learning with function approximations.

Example 1: Reproducing Kernel Reward Function. One popular option to parameterize the reward by functions is the reproducing kernel Hilbert space (RKHS, Kim and Park (2018); Li et al. (2018)). There have been several implementations of RKHS, and we consider the feature mapping approach. Specifically, we consider $g : \mathbb{R}^{d_{\mathcal{S}}} \times \mathbb{R}^{d_{\mathcal{A}}} \to \mathbb{R}^q$, and the reward can be written as $r(s,a) = \widetilde{r}_\theta(\psi_s, \psi_a) = \theta^\top g(\psi_s, \psi_a)$, where $\theta \in \mathbb{R}^q$. We require g to be Lipschitz continuous with respect to (ψ_a, ψ_s).

Assumption 3. The feature mapping g satisfies $g(0,0) = 0$, and there exists a constant ρ_g such that for any ψ_a, ψ_a', ψ_s and ψ_s', we have $\|g(\psi_s, \psi_a) - g(\psi_s', \psi_a')\|_2^2 \le \rho_g \sqrt{\|\psi_s - \psi_s'\|_2^2 + \|\psi_a - \psi_a'\|_2^2}$.

Assumption 3 is mild and satisfied by popular feature mappings, e.g., random Fourier feature mapping[1] (Rahimi and Recht, 2008; Bach, 2017). The next corollary presents the generalization bound of GAIL using feature mapping.

Corollary 1. Suppose $\|\theta\|_2 \le B_\theta$. For large enough n and T, with probability at least $1-\delta$ over the joint distribution of $\{(a_t^i, s_t^i)_{t=0}^{T-1}\}_{i=1}^n$, we have

$$d_{\mathcal{R}}(\pi^*, \widehat{\pi}) - \inf_\pi d_{\mathcal{R}}(\pi^*, \pi) \le O\left(\frac{\rho_g B_\theta}{\sqrt{nT/\zeta}} \sqrt{q \log\left(\rho_g B_\theta \sqrt{nT/\zeta}\right)} + \rho_g B_\theta \sqrt{\frac{\log(1/\delta)}{nT/\zeta}} \right) + \epsilon.$$

Corollary 1 indicates that with respect to a class of properly normalized reproducing kernel reward functions, GAIL generalizes in terms of the $\mathcal{R}$-distance.

Example 2: Neural Network Reward Function. Another popular option to parameterize the reward function is to use neural networks. Specifically, let $\sigma(v) = [\max\{v_1, 0\}, ..., \max\{v_d, 0\}]^\top$ denote the ReLU activation for $v \in \mathbb{R}^d$. We consider a D-layer feedforward neural network with ReLU activation as follows,

$$r(s,a) = \widetilde{r}_{\mathcal{W}}(\psi_s, \psi_a) = W_D^\top \sigma(W_{D-1} \sigma(...\sigma(W_1 [\psi_a^\top, \psi_s^\top]^\top))),$$

where $\mathcal{W} = \{W_i \mid W_i \in \mathbb{R}^{d_{i-1} \times d_i},\ i = 1, ..., D-1,\ W_D \in \mathbb{R}^{d_{D-1}}\}$ and $d_0 = d_{\mathcal{A}} + d_{\mathcal{S}}$. The next corollary presents the generalization bound of GAIL using neural networks.

Corollary 2. Suppose $\|W_i\|_2 \le 1$, where $i = 1, ..., D$. For large enough n and T, with probability at least $1-\delta$ over the joint distribution of $\{(a_t, s_t)_{t=0}^{T-1}\}_{i=1}^n$, we have

$$d_{\mathcal{R}}(\pi^*, \widehat{\pi}) - \inf_\pi d_{\mathcal{R}}(\pi^*, \pi) \le O\left(\frac{1}{\sqrt{nT/\zeta}} \sqrt{d^2 D \log\left(D\sqrt{dnT/\zeta}\right)} + \sqrt{\frac{\log(1/\delta)}{nT/\zeta}} \right) + \epsilon.$$

3 Computation of GAIL

To investigate the computational properties of GAIL, we parameterize the reward by functions belonging to some reproducing kernel Hilbert space. The policy can be parameterized by functions belonging to some reproducing kernel Hilbert space or some class of deep neural networks with

[1] More precisely, Assumption 3 actually holds with overwhelming probability over the distribution of the random mapping.

parameter ω. Specifically, we denote $\pi(a|s) = \widetilde{\pi}_\omega(\psi_s)$, where $\widetilde{\pi}_\omega(\psi_s)$ is the parametrized policy mapping from $\mathbb{R}^{d_S}$ to a simplex in $\mathbb{R}^d_{\mathcal{A}}$ with $|\mathcal{A}| = d$. For computational convenience, we consider solving a slightly modified minimax optimization problem:

$$\min_\omega \max_{\|\theta\|_2 \le \kappa} \mathbb{E}_{\widetilde{\pi}_\omega}[\widetilde{r}_\theta(s,a)] - \mathbb{E}_{\pi^*}\widetilde{r}_\theta[(\psi_s,\psi_a)] - \lambda H(\widetilde{\pi}_\omega) - \tfrac{\mu}{2}\|\theta\|_2^2, \tag{2}$$

where $\widetilde{r}_\theta(s,a) = \theta^\top g(\psi_s,\psi_a)$, $H(\widetilde{\pi}_\omega)$ is some regularizer for the policy (e.g., causal entropy regularizer, Ho and Ermon (2016)), and $\lambda > 0$ and $\mu > 0$ are tuning parameters.

We apply the alternating mini-batch stochastic gradient algorithm to (2). Specifically, we denote the objective function in (2) as $F(\omega,\theta)$ for notational simplicity. At the $(t+1)$-th iteration, we take

$$\theta^{(t+1)} = \Pi_\kappa\big(\theta^{(t)} + \tfrac{\eta_\theta}{q_\theta}\textstyle\sum_{j\in\mathcal{M}_\theta^{(t)}} \nabla_\theta f_j(\omega^{(t)},\theta^{(t)})\big) \quad \text{and} \tag{3}$$

$$\omega^{(t+1)} = \omega^{(t)} - \tfrac{\eta_\omega}{q_\omega}\textstyle\sum_{j\in\mathcal{M}_\omega^{(t)}} \nabla_\omega \widetilde{f}_j(\omega^{(t)},\theta^{(t+1)}), \tag{4}$$

where η_θ and η_ω are learning rates, the projection $\Pi_\kappa(v) = \mathbb{1}(\|v\|_2 \le \kappa)\cdot v + \mathbb{1}(\|v\|_2 > \kappa)\cdot\kappa\cdot v/\|v\|_2$, ∇f_j's and $\nabla\widetilde{f}_j$'s are independent stochastic approximations of ∇F (Sutton et al., 2000), and $\mathcal{M}_\theta^{(t)}$, $\mathcal{M}_\omega^{(t)}$ are mini-batches with sizes q_θ and q_ω, respectively. Before we proceed with the convergence analysis, we impose the follow assumptions on the problem.

Assumption 4. There are two positive constants M_ω and M_θ such that for any ω and $\|\theta\|_2 \le \kappa$,

$$\text{Unbiased}: \ \mathbb{E}\nabla f_j(\omega,\theta) = \mathbb{E}\nabla\widetilde{f}_j(\omega,\theta) = \nabla F(\omega,\theta),$$

$$\text{Bounded}: \ \mathbb{E}\|\nabla_\omega\widetilde{f}_j(\omega,\theta) - \nabla_\omega F(\omega,\theta)\|_2^2 \le M_\omega \quad \text{and} \quad \mathbb{E}\|\nabla_\theta f_j(\omega,\theta) - \nabla_\theta F(\omega,\theta)\|_2^2 \le M_\theta.$$

Assumption 4 requires the stochastic gradient to be unbiased with a bounded variance, which is a common assumption in existing optimization literature (Nemirovski et al., 2009; Ghadimi and Lan, 2013; Duchi et al., 2011; Bottou, 2010).

Assumption 5. (i) For any ω, there exists some constant $\chi > 0$ and $\upsilon \in (0,1)$ such that $\|(P_{\widetilde{\pi}_\omega})^t\rho_0 - \rho_{\widetilde{\pi}_\omega}\|_{\text{TV}} \le \chi\upsilon^t$, where $P_{\widetilde{\pi}_\omega}(s',a' \mid s,a) = \widetilde{\pi}_\omega(a'|s')P(s' \mid s,a)$ is the transition kernel induced by $\widetilde{\pi}_\omega$, ρ_0 is the initial distribution of (s_0,a_0), and $\rho_{\widetilde{\pi}_\omega}$ is the stationary distribution induced by $\widetilde{\pi}_\omega$.

(ii) There exist constants $S_{\widetilde{\pi}}, B_\omega, L_\rho, L_Q > 0$ such that for any ω,ω', we have

$$\|\nabla_\omega\log(\widetilde{\pi}_\omega(a|s)) - \nabla_\omega\log(\widetilde{\pi}_{\omega'}(a|s))\|_2 \le S_{\widetilde{\pi}}\|\omega-\omega'\|_2, \quad \|\nabla_\omega\log\widetilde{\pi}_\omega(a|s)\|_2 \le B_\omega,$$

$$\|\rho_{\widetilde{\pi}_\omega} - \rho_{\widetilde{\pi}'_\omega}\|_{\text{TV}} \le L_\rho\|\omega-\omega'\|_2, \quad \|Q^{\widetilde{\pi}_\omega} - Q^{\widetilde{\pi}_{\omega'}}\|_\infty \le L_Q\|\omega-\omega'\|_2,$$

where $Q^{\widetilde{\pi}_\omega}(s,a) = \sum_{t=0}^\infty \mathbb{E}\left[\widetilde{r}(s_t,a_t) - \mathbb{E}_{\widetilde{\pi}_\omega}[\widetilde{r}] \mid s_0 = s, a_0 = a, \widetilde{\pi}_\omega\right]$ is the action-value function.

(iii) There exist constants B_H and $S_H > 0$ such that for any ω,ω', we have

$$H(\widetilde{\pi}_\omega) \le B_H, \quad \text{and} \quad \|\nabla_\omega H(\widetilde{\pi}_\omega) - \nabla_\omega H(\widetilde{\pi}_{\omega'})\|_2 \le S_H\|\omega-\omega'\|_2.$$

Note that (i) of Assumption 5 requires the Markov Chain to be geometrically mixing. (ii) and (iii) state some commonly used regularity conditions for policies (Sutton et al., 2000; Pirotta et al., 2015).

We then define L-stationary points of F. Specifically, we say that (ω^*,θ^*) is a stationary point of F, if and only if, for any fixed $\alpha > 0$, $\nabla_\omega F(\omega^*,\theta^*) = 0$ and $\theta^* - \Pi_\kappa(\theta^* + \alpha\nabla_\theta F(\omega^*,\theta^*)) = 0$. Accordingly, we take $\alpha = 1$ and measure the sub-stationarity of the algorithm at the iteration N by

$$J_N = \min_{1\le t\le N}\mathbb{E}\|\theta^{(t)} - \Pi_\kappa(\theta^{(t)} + \nabla_\theta F(\omega^{(t)},\theta^{(t)}))\|_2^2 + \mathbb{E}\|\nabla_\omega F(\omega^{(t)},\theta^{(t+1)})\|_2^2.$$

We then state the global convergence of the alternating mini-batch stochastic gradient algorithm.

Theorem 2. Suppose Assumptions 1-5 hold. We choose step sizes η_θ, η_ω satisfying

$$\eta_\omega \le \min\left\{\frac{L_\omega}{S_\omega(8L_\omega+2)}, \frac{1}{2L_\omega}\right\}, \quad \eta_\theta \le \min\left\{\frac{1}{150\mu}, \frac{7L_\omega+1}{150S_\omega^2}, \frac{1}{100(2\mu+S_\omega)}\right\},$$

and meanwhile $\eta_\omega/\eta_\theta \le \mu/(30L_\omega+5)$, where $L_\omega = 2\sqrt{2}(S_{\widetilde{\pi}} + 2B_\omega L_\rho)\kappa\rho_g\chi/(1-\upsilon) + B_\omega L_Q$, and $S_\omega = 2\sqrt{2q}\kappa\rho_g\chi B_\omega/(1-\upsilon)$. Given any $\epsilon > 0$, we choose batch sizes $q_\theta = \widetilde{O}(1/\epsilon)$ and $q_\omega = \widetilde{O}(1/\epsilon)$. Then we need at most

$$N = \eta(C_0 + 4\sqrt{2}\rho_g\kappa + \mu\kappa^2 + 2\lambda B_H)\epsilon^{-1}$$

iterations such that $J_N \le \epsilon$, where C_0 depends on the initialization, and η depends on η_ω and η_θ.

4 Experiment

To verify our theory in Section 3, we conduct experiments in three reinforcement learning tasks: Acrobot, MountainCar, and Hopper. For each task, we first train an expert policy using the proximal policy optimization (PPO) algorithm in (Schulman et al., 2017) for 500 iterations, and then use the expert policy to generate the demonstration data. The demonstration data for every task contains 500 trajectories, each of which is a series of state action pairs throughout one episode in the environment. When training GAIL, we randomly select a mini-batch of trajectories, which contain at least 8196 state action pairs. We use PPO to update the policy parameters. This avoids the instability of the policy gradient algorithm, and improves the reproducibility of our experiments.

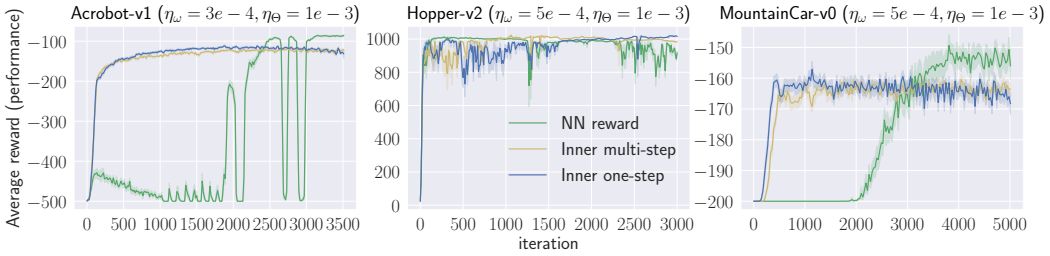

Figure 1: Performance of GAIL on three different tasks. The plotted curves are averaged over 5 independent runs with the vertical axis being the average reward and horizontal axis being the number of iterations.

We use the same neural network architecture for all the environments. For policy, we use a fully connected neural network with two hidden layers of 128 neurons in each layer and $\tanh$ activation. For reward, we use a fully connected ReLU neural network with two hidden layers of 1024 and 512 neurons, respectively. To implement the kernel reward, we fix the first two layers of the neural network after random initialization and only update the third layer, i.e., the first two layers mimic the random feature mapping. We choose $\kappa = 1$ and $\mu = 0.3$. When updating the neural network reward, we use weight normalization in each layer (Salimans and Kingma, 2016).

When updating the kernel reward at each iteration, we choose to take the stochastic gradient ascent step for either once (i.e., alternating update in Section 3) or 10 times. When updating the neural network reward at each iteration, we choose to take the stochastic gradient ascent step for only once. We tune step size parameters for updating the policy and reward, and summarize the numerical results of the step sizes attaining the maximal average episode reward in Figure 1.

As can be seen, using multiple stochastic gradient ascent steps for updating the reward at each iteration yields similar performance as that of one step. We present the convergence analysis of using multiple stochastic gradient ascent steps for updating the reward in Appendix C. Moreover, we observe that parameterizing the reward by neural networks slightly outperform that of the kernel reward. However, its training process tends to be unstable and takes longer time to converge.

5 Discussions

Our proposed theories of GAIL are closely related to Generative Adversarial Networks (Goodfellow et al., 2014; Arjovsky et al., 2017): (1) The generalization of GANs is defined based on the integral probabilistic metric (IPM) between the synthetic distribution obtained by the generator network and the distribution of the real data (Arora et al., 2017). As the real data in GANs are considered as independent realizations of the underlying distribution, the generalization of GANs can be analyzed using commonly used empirical process techniques for i.i.d. random variables. GAIL, however, involves dependent demonstration data from experts, and therefore the analysis is more involved. (2) Our computational theory of GAIL can be applied to MMD-GAN and its variants, where the IPM is induced by some reproducing kernel Hilbert space (Li et al., 2017; Bińkowski et al., 2018; Arbel et al., 2018). The alternating mini-batch stochastic gradient algorithm attains a similar sublinear rate of convergence to a stationary solution.

Moreover, our computational theory of GAIL only considers the policy gradient update when learning the policy (Sutton et al., 2000). Extending to other types of updates such as natural policy gradient (Kakade, 2002), proximal policy gradient (Schulman et al., 2017) and trust region policy optimization (Schulman et al., 2015) is a challenging, but important future direction.

References

ABBEEL, P. and NG, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM.

ABOUNADI, J., BERTSEKAS, D. and BORKAR, V. S. (2001). Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, **40** 681–698.

ANTHONY, M. and BARTLETT, P. L. (2009). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.

ARBEL, M., SUTHERLAND, D., BIŃKOWSKI, M. and GRETTON, A. (2018). On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*.

ARGALL, B. D., CHERNOVA, S., VELOSO, M. and BROWNING, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, **57** 469–483.

ARJOVSKY, M., CHINTALA, S. and BOTTOU, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning*.

ARORA, S., GE, R., LIANG, Y., MA, T. and ZHANG, Y. (2017). Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

BACH, F. (2017). On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, **18** 714–751.

BEN-TAL, A. and NEMIROVSKI, A. (1998). Robust convex optimization. *Mathematics of operations research*, **23** 769–805.

BIŃKOWSKI, M., SUTHERLAND, D. J., ARBEL, M. and GRETTON, A. (2018). Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

BOTTOU, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

BRAFMAN, R. I. and TENNENHOLTZ, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, **3** 213–231.

CAI, Q., HONG, M., CHEN, Y. and WANG, Z. (2019). On the global convergence of imitation learning: A case for linear quadratic regulator. *arXiv preprint arXiv:1901.03674*.

CHAMBOLLE, A. and POCK, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, **40** 120–145.

CHEN, Y., LAN, G. and OUYANG, Y. (2014). Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, **24** 1779–1814.

DUCHI, J., HAZAN, E. and SINGER, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, **12** 2121–2159.

FINN, C., LEVINE, S. and ABBEEL, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*.

GHADIMI, S. and LAN, G. (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, **23** 2341–2368.

GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. and BENGIO, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*.

HO, J. and ERMON, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*.

HOWARD, R. A. (1960). Dynamic programming and markov processes.

KAELBLING, L. P., LITTMAN, M. L. and MOORE, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, **4** 237–285.

KAKADE, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems*.

KEARNS, M. and SINGH, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine learning*, **49** 209–232.

KIM, K.-E. and PARK, H. S. (2018). Imitation learning via kernel mean embedding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

KUEFLER, A., MORTON, J., WHEELER, T. and KOCHENDERFER, M. (2017). Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE.

LECUN, Y., BENGIO, Y. and HINTON, G. (2015). Deep learning. *nature*, **521** 436.

LEVIN, D. A. and PERES, Y. (2017). *Markov chains and mixing times*, vol. 107. American Mathematical Soc.

LEVINE, S. and KOLTUN, V. (2012). Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617*.

LI, C.-L., CHANG, W.-C., CHENG, Y., YANG, Y. and PÓCZOS, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*.

LI, L., LITTMAN, M. L., WALSH, T. J. and STREHL, A. L. (2011). Knows what it knows: a framework for self-aware learning. *Machine learning*, **82** 399–443.

LI, S., XIAO, S., ZHU, S., DU, N., XIE, Y. and SONG, L. (2018). Learning temporal point processes via reinforcement learning. In *Advances in Neural Information Processing Systems*.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A. K., OSTROVSKI, G. ET AL. (2015). Human-level control through deep reinforcement learning. *Nature*, **518** 529.

MOHRI, M., ROSTAMIZADEH, A. and TALWALKAR, A. (2018). *Foundations of machine learning*. MIT press.

MÜLLER, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, **29** 429–443.

MURRAY, W. and OVERTON, M. L. (1980). A projected lagrangian algorithm for nonlinear minimax optimization. *SIAM Journal on Scientific and Statistical Computing*, **1** 345–370.

NEMIROVSKI, A., JUDITSKY, A., LAN, G. and SHAPIRO, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, **19** 1574–1609.

NG, A. Y., RUSSELL, S. J. ET AL. (2000). Algorithms for inverse reinforcement learning. In *Icml*, vol. 1.

ORMONEIT, D. and SEN, Ś. (2002). Kernel-based reinforcement learning. *Machine learning*, **49** 161–178.

PFEIFFER, M., SHUKLA, S., TURCHETTA, M., CADENA, C., KRAUSE, A., SIEGWART, R. and NIETO, J. (2018). Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. *IEEE Robotics and Automation Letters*, **3** 4423–4430.

PIROTTA, M., RESTELLI, M. and BASCETTA, L. (2015). Policy gradient in lipschitz markov decision processes. *Machine Learning*, **100** 255–283.

POMERLEAU, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, **3** 88–97.

PUTERMAN, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

RAHIMI, A. and RECHT, B. (2008). Random features for large-scale kernel machines. In *Advances in neural information processing systems*.

RUSSELL, S. J. (1998). Learning agents for uncertain environments. In *COLT*, vol. 98.

SALIMANS, T. and KINGMA, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*.

SCHULMAN, J., LEVINE, S., ABBEEL, P., JORDAN, M. and MORITZ, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*.

SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A. and KLIMOV, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

STREHL, A. L. and LITTMAN, M. L. (2005). A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*. ACM.

SUTTON, R. S., BARTO, A. G. ET AL. (1998). *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge.

SUTTON, R. S., MCALLESTER, D. A., SINGH, S. P. and MANSOUR, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*.

SYED, U., BOWLING, M. and SCHAPIRE, R. E. (2008). Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*. ACM.

TAIL, L., ZHANG, J., LIU, M. and BURGARD, W. (2018). Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

VALLENDER, S. (1974). Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, **18** 784–786.

VAPNIK, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

WILLEM, M. (1997). *Minimax theorems*, vol. 24. Springer Science & Business Media.

YU, B. (1994). Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability* 94–116.