Imitation Learning: An Introduction

Ziniu Li

The Chinese University of Hong Kong, Shenzhen

Overview

What to expect from this talk?

- Imitation Learning Algorithms (e.g., BC and GAIL)

- Algorithmic analysis (e.g., generalization and sample complexity)

Table of contents

1. What is Imitation Learning?

2. Behavioral Cloning

3. Adversarial Imitation Learning

What is Imitation Learning?

What is Imitation Learning?

Imitation learning (a.k.a., learning from demonstrations)

- "... efficiently learn a desired behavior by imitating an expert's behavior" [Osa et al., 2018].
- "... aim to mimic human behavior in a given task" [Hussein et al., 2017].

Indeed, any desired behavior could be viewed the expert's behavior.

What is Imitation Learning?

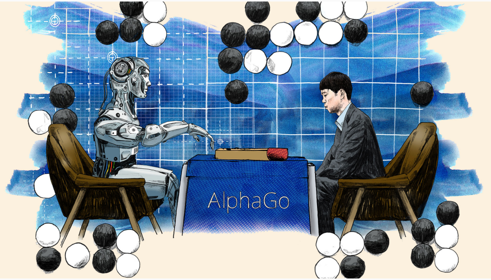


Figure 1: AlphaGo: the agent imitates to play the game Go [Silver et al., 2016]. Photo is from the finical times (Internet).

What is Imitation Learning?

Figure 2: Music generation: the agent learns to generate polyphonic music. Figure is from [Lee et al., 2017].

What is Imitation Learning?

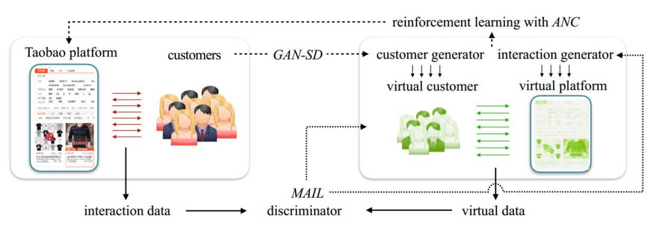


Figure 3: Virtual Taobao: the agent learns to recover the true real-world environment. Figure is from [Shi et al., 2019].

Abstract Model for Imitation Learning

Markov Decision Processes (MDPs): $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \rho, H)$

- (Finite) state space $\mathcal{S}$ and action space $\mathcal{A}$.
- Transition function $P(s'|s, a)$ specifies the probability of going to s' conditioned (s, a).
- $R(s, a)$ determines the reward.
- ρ is the initial state distribution.
- H is the maximum length of a trajectory.

Three important concepts:

- Policy $\pi(a|s)$ determines how to interact with the environment.
- Return $V(\pi) = \mathbb{E}[\sum_{h=1}^{H} r(s_h, a_h)]$ determines the value.
- State-action distribution $d_h^{\pi}(s, a) := \mathbb{P}(s_h = s, a_h = a)$ determines the visiting probability of (s, a) in time step h.

Abstract Model for Imitation Learning

Task description: Given a dataset that records the expert trajectories, we need to learn a policy that matches the expert performance.

$$\text{expert trajectory :} \mathrm{tr} = \left\{(s_1, a_1), \dots, (s_H, a_H)\right\},$$
$$\text{expert dataset :} \mathcal{D} = \left\{\mathrm{tr}^1, \cdots, \mathrm{tr}^N\right\}$$

We measure the performance of imitation learning algorithms by

$$\text{imitation gap : } V(\pi) - V(\pi^{\mathsf{E}})$$

where $V(\pi)$ is the long-term return of a policy π.

Connection with Supervised Learning

Comparison between Imitation Learning and Supervised Learning.

Similarity:

- Dataset contains the input signal (i.e., state) and the output signal (i.e., action).
- Aim to learn a mapping from the input to the output.

Difference:

- Supervised learning assumes that samples are i.i.d from an unknown distribution μ, i.e., $(s, a) \sim \mu$.
- State-action pairs in imitation learning are correlated, i.e., $s_{t+1} = P(\cdot|s_t, a_t)$.

Behavioral Cloning

Behavioral Cloning

Main Idea: maximum likelihood estimation for π^{E}:

$$\max_{\pi} \sum_{h=1}^{H} \sum_{(s,a)\in \mathrm{tr}_h} \log \pi_h(a|s)$$

Optimal solution: in the tabular set, we have

$$\pi_h^{\mathsf{BC}}(a|s) = \begin{cases} \frac{\#\mathrm{tr}_h(\cdot,\cdot)=(s,a)}{\sum_{a'} \#\mathrm{tr}_h(\cdot,\cdot)=(s,a')} & \text{if } \sum_{a'} \#\mathrm{tr}_h(\cdot,\cdot) = (s,a') > 0 \\ 1/|\mathcal{A}| & \text{otherwise} \end{cases} \tag{1}$$

That is, $\pi_h^{\mathsf{BC}}(a|s)$ counts how many times action a appeared.

Behavioral Cloning

Algorithmic Behaviors of BC:

- $\pi_h^{\mathrm{BC}}(a|s)$ is well defined if s appeared in the dataset. For a non-visited state s, $\pi_h^{\mathrm{BC}}(\cdot|s)$ is a random policy. Thus, π^{BC} suffers the optimality gap from non-visited states. In the worst-case, the optimality gap could be H.

- If the number of expert trajectories goes to infinity, the number of non-visited states diminishes to zero, and therefore π_h^{BC} can perfectly recover π_h^{E}.

- Since BC only requires a fixed expert dataset, it is a standard offline algorithm.

Behavioral Cloning

Theorem 1 Sample Complexity Bound for BC [Rajaraman et al., 2020]

Consider tabular and episodic MDPs. Assume that agent has access to N expert trajectories of length H collected by a deterministic π^{E} and implements the BC's policy as in Eq.(1). Then, in expectation, we have

$$\mathbb{E}\left[V(\pi^{\mathsf{BC}})\right] - V(\pi^{\mathsf{E}}) \precsim \frac{|\mathcal{S}|H^2}{N},$$

where the expectation is taken over the randomness in the dataset collection.

In the stochastic expert case, the sub-optimality has an extra dependence on $\log(N)$.

Lower Bound of Offline Imitation Learning

We need to check whether Theorem 1 is tight or not.

Theorem 2 Lower Bound of Offline Imitation Learning [Rajaraman et al., 2020]

Consider tabular and episodic MDPs. Assume that agent has access to N expert trajectories of length H collected by a deterministic π^{E}. For **any** offline imitation learning algorithm, the imitation gap of its output policy is at least

$$\mathbb{E}\left[V(\pi)\right] - V(\pi^{\mathsf{E}}) \gtrsim \frac{|\mathcal{S}|H^2}{N},$$

where the expectation is taken over the randomness in the dataset collection.

We see that BC is minimax optimal in the offline setting.

Lower Bound of Offline Imitation Learning

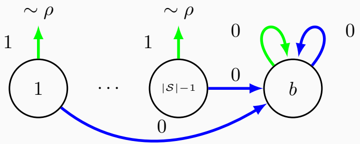
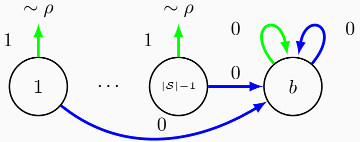


Figure 4: Lower bound instances (named **Reset Cliff**) in the offline imitation setting [Rajaraman et al., 2020].

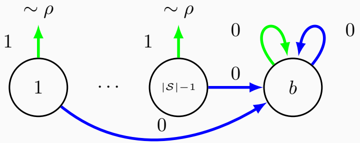
- The first $|\mathcal{S}| - 1$ states are good and non-absorbing, whereas the last state "*b*" is bad and absorbing.
- The green action (with reward 1) resets the transition according to the initial state distribution ρ, whereas the blue action (with reward 0) leads the agent to the absorbing state.
- The expert policy always takes the green action, and ρ has the support only on the first $|\mathcal{S}| - 1$ states.

Lower Bound of Offline Imitation Learning



- Given N expert trajectories, if the state space is large, i.e., $|\mathcal{S}| > N$, it is likely that not all good states are visited and covered.
- For any offline algorithm, it cannot infer the expert action on a non-visited state.
- In this case, it selects the wrong action with probability at least 0.5 and suffers an optimality gap at most H.

Active Learning May Not Address the Compounding Errors Issue



Let us consider an the active learning setting where the agent can interact with the environment to query expert actions (like the setting in DAgger's paper [Ross et al., 2011]).

- Unfortunately, Rajaraman et al. [2020] showed that once the agent makes a wrong decision, it can only collect the information from the absorbing state.
- This kind of dataset is uninformative as an offline dataset. Thus, algorithms like DAgger cannot address the compounding errors issue here.

Historical Remark

- BC algorithm dates back to [Pomerleau, 1991], in which an autonomous car was trained.
- [Syed and Schapire, 2010, Ross et al., 2011] provided the first analysis to argue that BC may suffer **“compounding errors”**, which corresponds to the bad absorbing state in the lower bound instances. This type of analysis is extended in [Xu et al., 2020, Swamy et al., 2021].
- The above analysis mainly holds in a population level, and the finite-samples complexity bound is shown in [Rajaraman et al., 2020, Xu et al., 2020].
- As mentioned, the analysis for BC can be extended to analyze the classical model-based RL algorithms, in which the transition function is imitated from samples; see [Xu et al., 2020].

Historical Remark

Applications of BC.

- **Recovering an expert policy:**
  Autonomous driving [Codevilla et al., 2018], mobile robotics [Giusti et al., 2016], opponent modeling in MARL [Foerster et al., 2018].

- **Pretraining of an RL algorithm:**
  Learning an initial policy from expert demonstrations [Silver et al., 2016, Vinyals et al., 2019, Hester et al., 2018],

- **Environment virtualization:**
  Learning MuJoCo simulator [Wang et al., 2019]

Adversarial Imitation Learning

Open Question of Adversarial Imitation Learning

Adversarial imitation learning (AIL) methods become popular since GAIL (generative adversarial imitation learning) [Ho and Ermon, 2016].

- GAIL is reported to beat BC for many MuJoCo locomotion tasks.

- For MuJoCo tasks, the imitation gap of GAIL is almost zero, even when only 1 expert trajectory is provided.

- GAIL is not an offline algorithm, as it requires the environment interaction.

Open Question of Adversarial Imitation Learning

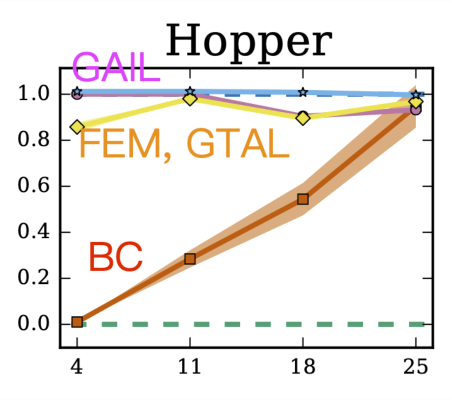


Figure 5: Performance of BC, GAIL, and other AIL methods (FEM and GTAL) on Hopper (a MuJoCo task) [Ho and Ermon, 2016]. y-axis is the (expert) normalized score.

Open question: **why is GAIL much better than BC especially in the low-data regime?**

Generative Adversarial Imitation Learning

Adversarial formulation of GAIL:

$$\min_{\pi}\max_{c}\sum_{h=1}^{H}\mathbb{E}_{(s,a)\sim d_h^{\pi}}\left[\log c_h(s,a)\right]+\mathbb{E}_{(s,a)\sim d_h^{\pi^{\mathsf{E}}}}\left[\log(1-c(s,a))\right]$$

where c is the called the discriminator and its optimal solution gives the following state-action distribution matching formulation:

$$\min_{\pi}\sum_{h=1}^{H}D_{\mathrm{JS}}(d_h^{\pi},d_h^{\pi^{\mathsf{E}}})-\lambda H(\pi_h),$$

where

$$D_{\mathrm{JS}}(P,Q)=D_{\mathrm{KL}}(P,(P+Q)/2)+D_{\mathrm{KL}}(Q,(P+Q)/2)$$

and $D_{\mathrm{KL}}(P,Q)=\sum_x p(x)\log(p(x)/q(x))$, and $H(P)=-\sum_x\log(p(x))$.

Adversarial Imitation Learning

We can generalize GAIL by the following state-action distribution matching form:

$$\min_{\pi} \sum_{h=1}^{H} \psi(d_h^{\pi}, d_h^{\pi^{\mathsf{E}}}),$$

where ψ is a distance metric. We require $\psi(P, Q) = 0$ if $P = Q$.

This formulation unifies the (so-called) adversarial imitation learning and apprenticeship learning algorithms.

In practice, $d_h^{\pi^{\mathsf{E}}}$ is often estimated from finite samples:

$$\widehat{d_h^{\pi^{\mathsf{E}}}} = \sum_{\mathrm{tr}_h} \frac{\#\mathrm{tr}_h(\cdot, \cdot) = (s, a)}{N}. \tag{2}$$

Adversarial Imitation Learning

Consider the practical formulation:

$$\min_{\pi} \sum_{h=1}^{H} \psi(d_h^{\pi}, \widehat{d_h^{\pi^{\mathrm{E}}}}), \tag{3}$$

- Optimizing Eq.(3) requires d_h^{π}, which further relies on the transition function P. Hence, AIL methods cannot be directly applied in the offline setting.
- For BC, its optimization problem is independent in each time step. Thus, BC involves a convex optimization problem.
- For AIL, its optimization in each time step is correlated by d_h^{π}. As a result, AIL involves a non-convex optimization problem.
- The optimization for non-visited states are well-defined for AIL.

AIL Can be Worse than BC

We analyze AIL in Eq.(3) by assuming a) the transition function is known; b) the global optimal solution to Eq.(3) is available; c) $\psi(P, Q) = \sum_x |p(x) - q(x)|$. We call such an method VAIL (vanilla AIL).

Theorem 3 Tight Sample Complexity Bound for VAIL [Xu et al., 2022]

Consider tabular and episodic MDPs. Assume that agent has access to N expert trajectories of length H collected by a deterministic π^{E} and implements the AIL method as in Eq.(3), and the mentioned assumptions hold. Then, in expectation, we have

$$\mathbb{E}\left[V(\pi^{\mathsf{VAIL}})\right] - V(\pi^{\mathsf{E}}) \approx H\sqrt{\frac{|\mathcal{S}|}{N}},$$

where the expectation is taken over the randomness in the dataset collection.

AIL Can be Worse than BC

Imitation gap:

$$\text{BC}: \quad \underbrace{\mathbb{E}\left[V(\pi^{\mathsf{BC}})\right] - V(\pi^{\mathsf{E}})}_{\varepsilon} \precsim \frac{|\mathcal{S}|H^2}{N} \quad \Longrightarrow \quad N \succsim \frac{|\mathcal{S}|H^2}{\varepsilon}$$

$$\text{VAIL}: \quad \underbrace{\mathbb{E}\left[V(\pi^{\mathsf{VAIL}})\right] - V(\pi^{\mathsf{E}})}_{\varepsilon} \approxeq H\sqrt{\frac{|\mathcal{S}|}{N}} \quad \Longrightarrow \quad N \approxeq \frac{|\mathcal{S}|H^2}{\varepsilon^2}$$

When the desired error $\varepsilon \in (0, 1)$, VAIL requires more expert trajectories than BC.

Other AIL methods [Abbeel and Ng, 2004, Syed and Schapire, 2007] with different choices of ψ have a similar upper bound $\mathcal{O}(H\sqrt{|\mathcal{S}|/N})$, so that the conclusion hold for other AIL methods.

AIL Can be Worse than BC

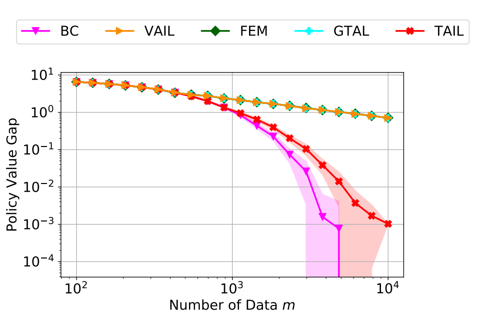


Figure 6: Performance of BC, VAIL and other AIL algorithms (FEM, GTAL, and TAIL) on Standard Imitation [Xu et al., 2022].

To achieve the same imitation gap, VAIL, FEM [Abbeel and Ng, 2004], and GTAL [Syed and Schapire, 2007] require more expert trajectories than BC. TAIL [Xu et al., 2022] is designed to overcome the sample barrier issue (explained later).

VAIL Can be Worse than BC

Standard Imitation: the hard instances for AIL.

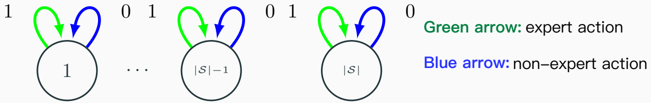


Figure 7: The hard instances for VAIL [Rajaraman et al., 2020, Xu et al., 2022].

- Each state is an absorbing state.
- Green action has a reward 1, and the blue action has reward 0.

Why VAIL Can be Worse than BC?

For simplicity, assume there are 2 states and $H = 1$. The initial state distribution $\rho = (0.5, 0.5)$.

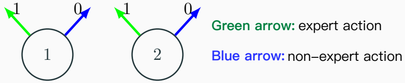


Figure 8: Simplified Standard Imitation.

Assume there are 10 expert trajectories.: 4 of them are from s^1 and 6 of them are from s^2.

BC exactly recovers the expert policy and the imitation gap is 0.

Why Can VAIL be Worse than BC?

Let a_1 be the green action and a_2 be the blue action. For AIL,

$$\widehat{d^{\pi^{\mathrm{E}}}}(s^1, a^1) = 0.4, \widehat{d^{\pi^{\mathrm{E}}}}(s^1, a^2) = 0.0,$$
$$\widehat{d^{\pi^{\mathrm{E}}}}(s^2, a^1) = 0.6, \widehat{d^{\pi^{\mathrm{E}}}}(s^2, a^2) = 0.0.$$

There are many globally optimal solutions. For instance, $\pi(a^1|s^1) = 0.8, \pi(a^2|s^1) = 0.2, \pi(a^1|s^2) = 1.0$, and

$$d^{\pi}(s^1, a^1) = 0.4, d^{\pi}(s^1, a^2) = 0.1,$$
$$d^{\pi}(s^2, a^1) = 0.5, d^{\pi}(s^2, a^2) = 0.0.$$

For such an optimal policy, the empirical loss is 0.2 and its imitation gap is 0.1, which is larger than BC.

Why Can VAIL be Worse than BC?

Intuition behind the poor performance of AIL [Xu et al., 2022]:

- Sample Barrier: The statistical estimation error (i.e., $\|d_h^{\pi^{\mathrm{E}}} - \widehat{d_h^{\pi^{\mathrm{E}}}}\|_1$) in AIL diminishes at a relatively slow rate $\mathcal{O}(\sqrt{|\mathcal{S}|/N})$.
- Weak Convergence: VAIL could make a wrong decision even on visited states from the expert demonstrations. In contrast, BC directly copies the expert action, which means BC never makes a mistake on a visited state.

Note that the bad performance of VAIL here does not contradict with the conclusion in GAIL because MuJoCo tasks are quite different from the hard instances here.

AIL Can be much better than BC

Back to the open question: is there an instance that AIL can be better than BC? The answer is yes!

Theorem 4 Horizon-free Sample Complexity Bound for VAIL [Xu et al., 2022]

Consider tabular and episodic MDPs. Assume that agent has access to N expert trajectories of length H collected by a deterministic π^{E} and implements the AIL method as in Eq.(3), and the mentioned assumptions hold. There exists a family of instances such that in expectation, we have

$$\mathbb{E}\left[V(\pi^{\mathsf{VAIL}})\right] - V(\pi^{\mathsf{E}}) \precsim \sqrt{\frac{|\mathcal{S}|}{N}},$$

where the expectation is taken over the randomness in the dataset collection.

AIL Can be much better than BC

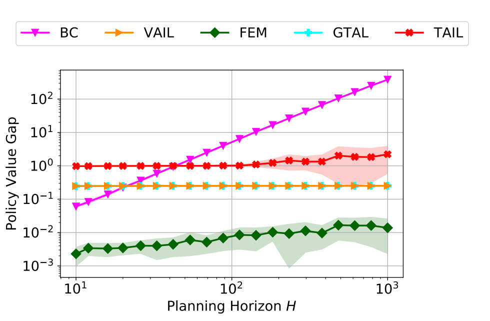


Figure 9: Performance of BC, VAIL, and other AIL methods (FEM, GTAL, TAIL) on Reset Cliff [Xu et al., 2022].

AIL methods can be much better than BC on long-horizon Reset Cliff tasks (i.e., the lower bound instances in the offline setting).

AIL Can be much better than BC

Consider a simple example. There are 3 states and 2 actions.

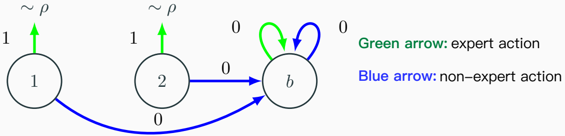


Figure 10: Simplified Reset Cliff.

The agent is provided only 2 expert trajectories:
$\mathrm{tr}^1 = (s^1, a^1) \to (s^1, a^1)$ and $\mathrm{tr}^2 = (s^1, a^1) \to (s^2, a^1)$.

For BC, it exactly recovers the expert action on visited states but it poses a uniform policy on the non-visited s^2 in time step $h = 1$.

As a result, BC makes a mistake with probability
$\rho(s^2) \cdot \pi_1^{\text{BC}}(a^2|s^2) = 0.25$, and its imitation gap is $0.25 \cdot 2 = 0.5$.

AIL Can be much better than BC

For VAIL, it make senses to guess that the expert action is recovered on visited states.

Claim: VAIL exactly recovers the expert action even on non-visited state s^2 in time step $h = 1$.

Assume $\pi_1(a^2|s^2) = 1 - \beta$, where $\beta \in [0, 1]$. Provided π takes the expert action elsewhere, we can calculate the loss function for β:

$$\text{Loss}(\beta) = \sum_{h=1}^{2} \sum_{(s,a)} |P_h^{\pi}(s, a) - \widehat{P}_h^{\pi^{\text{E}}}(s, a)| = 2 - \beta,$$

which has a unique globally optimal solution at $\beta = 1$.

This suggests that the claim is true. Consequently, the imitation gap of VAIL is 0, which is smaller than BC.

AIL Can be much better than BC

[Xu et al., 2022] showed that even given a single expert trajectory, AIL is able to recover the expert policy for Reset Cliff.

Theorem 4 implies that when the desired error $\varepsilon \geq (0, \mathcal{O}(1/H^2)]$ ($H = 10^3$ for MuJoCo tasks), VAIL is provably better than BC.

Since the instances in Theorem 4 are close to practical MuJoCo tasks, Theorem 4 explains the folklore that AIL is much better than BC.

To summarize, for some non-trivial instances, the state-action distribution matching provides effective guidance on non-visited states so that AIL methods perform well in practice.

Beyond Vanilla AIL

As mentioned, the vanilla AIL methods are not optimal in the worst-case because the estimation error concentrates at a relatively slow speed. Refer to Figure 6.

Rajaraman et al. [2020] presented an improved estimation for $d^{\pi^{\mathrm{E}}}$, and the resultant algorithm **MIMIC-MD** has the optimal worst-case bound in terms of H [Rajaraman et al., 2021].

This estimator explicitly leverages the transition function to improve the estimation, and this improvement is mainly observable in Standard Imitation type tasks.

Beyond Vanilla AIL

Decomposition of $d^{\pi^{\mathsf{E}}}$:

$$d_h^{\pi^{\mathsf{E}}}(s,a) = \sum_{\mathrm{tr}_h \in \mathsf{Tr}_h^{\mathcal{D}_1}} \mathbb{P}^{\pi^{\mathsf{E}}}(\mathrm{tr}_h)\mathbb{I}\left\{\mathrm{tr}_h(s_h,a_h)=(s,a)\right\} \tag{4a}$$

$$+ \sum_{\mathrm{tr}_h \notin \mathsf{Tr}_h^{\mathcal{D}_1}} \mathbb{P}^{\pi^{\mathsf{E}}}(\mathrm{tr}_h)\mathbb{I}\left\{\mathrm{tr}_h(s_h,a_h)=(s,a)\right\}. \tag{4b}$$

$\mathsf{Tr}_h^{\mathcal{D}_1}$: the trajectory set defined by a dataset $\mathcal{D}_1$.

- red part: the probability of visiting (s,a) by a trajectory $tr_h \in \mathsf{Tr}_h^{\mathcal{D}_1}$.
- blue part: the probability of visiting (s,a) by a trajectory $tr_h \notin \mathsf{Tr}_h^{\mathcal{D}_1}$.

Main idea: we can calculate the red part if we know the transition function and the dataset $\mathcal{D}_1$. That is, there is no estimation error.

Beyond Vanilla AIL

Improved estimator in [Rajaraman et al., 2020]:

$$\widetilde{d_h^{\pi^{\mathrm{E}}}}(s,a) = \sum_{\mathrm{tr}_h \in \mathsf{Tr}_h^{\mathcal{D}_1}} \mathbb{P}^{\pi^{\mathrm{E}}}(\mathrm{tr}_h)\mathbb{I}\left\{\mathrm{tr}_h(s_h,a_h) = (s,a)\right\} \tag{5a}$$

$$+ \frac{\sum_{\mathrm{tr}_h \in \mathcal{D}_1^c} \mathbb{I}\{\mathrm{tr}_h(s_h,a_h) = (s,a), \mathrm{tr}_h \notin \mathsf{Tr}_h^{\mathcal{D}_1}\}}{|\mathcal{D}_1^c|}. \tag{5b}$$

Split $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_1^c$ (a half cut), estimate the blue part in Eq.(4b) by leveraging $\mathcal{D}_1^c$ with an maximum likelihood estimation.

$$\text{VAIL}: \sum_{h=1}^{H} \left\| \widehat{d_h^{\pi^{\mathrm{E}}}} - d_h^{\pi^{\mathrm{E}}} \right\|_1 \precsim H\sqrt{\frac{|\mathcal{S}|}{N}}$$

$$\text{MIMIC-MD}: \sum_{h=1}^{H} \left\| \widetilde{d_h^{\pi^{\mathrm{E}}}} - d_h^{\pi^{\mathrm{E}}} \right\|_1 \precsim \min\left\{ H^{3/2}\frac{|\mathcal{S}|}{N}, H\sqrt{\frac{|\mathcal{S}|}{N}} \right\}.$$

MIMIC-MD has a smaller estimation error than VAIL.

Beyond Vanilla AIL

Computation Efficiency:

- The original MIMIC-MD algorithm is not polynomial solvable [Rajaraman et al., 2020].
- The linear programming formulation for MIMIC-MD is polynomial solvable in obtaining an exact solution but the space complexity is huge [Rajaraman et al., 2021].
- The adversarial formulation for MIMIC-MD (named TAIL) is also polynomial in solving an ε-approximate solution but its space complexity is cheap [Xu et al., 2022].

Practical Performance:

- MIMIC-MD and TAIL mainly outperform vanilla AIL methods on the Standard Imitation task (see Figure 6).
- All of them have a similar horizon-free performance on the Reset Cliff task (see Figure 9).

Historical Remark

It is recognized as an **open question** that why AIL performs significantly better than BC, especially in the low-data regime [Ghasemipour et al., 2019, Orsini et al.].

There are many efforts in explaining the generalization performance of GAIL [Wang et al., 2020, Xu et al., 2020, Zhang et al., 2020, Rajaraman et al., 2020, Liu et al., 2021].

In particular, [Xu et al., 2022] is the first to explain that AIL can achieve horizon-free sample complexity for some non-trivial instances.

Notably, [Rajaraman et al., 2020] made a remarkable step in improving the minimax optimality.

Historical Remark

Both AIL [Ho and Ermon, 2016, Fu et al., 2018] and apprenticeship learning algorithms [Abbeel and Ng, 2004, Syed and Schapire, 2007] implement state-action distribution matching, so they perform similarly in the tabular setting.

In the deep case, it is more easier to incorporate the feature learning in AIL methods.

Due to the good performance of AIL methods, such methods have been popular in real applications; see [Shi et al., 2019, Shang et al., 2019, Huzhang et al., 2021].

Online AIL

Let us know consider the **online** setting, in which the transition function is unknown but the interaction is allowed. GAIL is **interaction-inefficient** (e.g., 3M interaction steps for MuJoCo tasks).

From the empirical side, the interaction efficiency is improved by advanced off-policy RL algorithms. For instance, DAC [Kostrikov et al., 2019] utilized the TD3 algorithm [Fujimoto et al., 2018] to improve the interaction efficiency.

From the theoretical side, the interaction efficiency is improved by more advanced exploration strategies such as the reward-free exploration strategies [Xu et al., 2022] and no-regret algorithms [Shani et al., 2022].

Offline AIL

Recently, **offline AIL** methods are developed and reported results suggest that they can perform better than (the variants) of BC in the offline setting [Kostrikov et al., 2020, Sun et al., 2021, Kim et al., 2022b,a, Ma et al., 2022].

There results seem to contradict the lower bound in the offline setting. [Li et al., 2022] explained the confusing parts here.

- In the tabular case, offline AIL algorithm ValueDICE [Kostrikov et al., 2020] can reduce to BC in the offline setting.
- The good performance of ValueDICE relies on its regularization. With a simple regularization, BC's performance is comparative with ValueDICE.

The underlying reason why offline AIL is on-pair with BC is: the policy optimization is only defined over visited states in the offline setting, which is different from the online setting.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning*, volume 69, 2004.

F. Codevilla, M. Müller, A. M. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation*, pages 1–9, 2018.

J. N. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130, 2018.

J. Fu, K. Luo, and S. Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1582–1591, 2018.

S. K. S. Ghasemipour, R. S. Zemel, and S. Gu. A divergence minimization perspective on imitation learning methods. In *Proceedings of the 3rd Annual Conference on Robot Learning*, pages 1259–1277, 2019.

References ii

A. Giusti, J. Guzzi, D. C. Ciresan, F. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.

T. Hester, M. Vecerík, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. P. Agapiou, J. Z. Leibo, and A. Gruslys. Deep q-learning from demonstrations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 3223–3230, 2018.

J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, pages 4565–4573, 2016.

A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50(2):1–35, 2017.

G. Huzhang, Z. Pang, Y. Gao, Y. Liu, W. Shen, W.-J. Zhou, Q. Da, A. Zeng, H. Yu, Y. Yu, et al. Aliexpress learning-to-rank: Maximizing online model performance without going online. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

G.-H. Kim, J. Lee, Y. Jang, H. Yang, and K.-E. Kim. Lobsdice: Offline imitation learning from observation via stationary distribution correction estimation. *arXiv*, 2202.13536, 2022a.

G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *Proceedings of the 10th International Conference on Learning Representations*, 2022b.

References iii

I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

S.-g. Lee, U. Hwang, S. Min, and S. Yoon. Polyphonic music generation with sequence generative adversarial networks. *arXiv*, 1710.11418, 2017.

Z. Li, T. Xu, Y. Yu, and Z.-Q. Luo. Rethinking valuedice - does it really improve performance? In *ICLR Blog Track*, 2022. https://iclr-blog-track.github.io/2022/03/25/rethinking-valuedice/.

Z. Liu, Y. Zhang, Z. Fu, Z. Yang, and Z. Wang. Provably efficient generative adversarial imitation learning for online and offline setting with linear function approximation. *arXiv*, 2108.08765, 2021.

Y. J. Ma, A. Shen, D. Jayaraman, and O. Bastani. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv*, 2202.02433, 2022.

M. Orsini, A. Raichuk, L. Hussenot, D. Vincent, R. Dadashi, S. Girgin, M. Geist, O. Bachem, O. Pietquin, and M. Andrychowicz. What matters for adversarial imitation learning? In *Advances in Neural Information Processing Systems 34*, pages 14656–14668.

T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotic*, 7(1-2):1–179, 2018.

References iv

D. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

N. Rajaraman, L. F. Yang, J. Jiao, and K. Ramchandran. Toward the fundamental limits of imitation learning. In *Advances in Neural Information Processing Systems 33*, pages 2914–2924, 2020.

N. Rajaraman, Y. Han, L. F. Yang, K. Ramchandran, and J. Jiao. Provably breaking the quadratic error compounding barrier in imitation learning, optimally. *arXiv*, 2102.12948, 2021.

S. Ross, G. J. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

W. Shang, Y. Yu, Q. Li, Z. T. Qin, Y. Meng, and J. Ye. Environment reconstruction with hidden confounders for reinforcement learning based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 566–576, 2019.

L. Shani, T. Zahavy, and S. Mannor. Online apprenticeship learning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 2022.

J. Shi, Y. Yu, Q. Da, S. Chen, and A. Zeng. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceeding of the 33rd AAAI Conference on Artificial Intelligence*, pages 4902–4909, 2019.

References v

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

M. Sun, A. Mahajan, K. Hofmann, and S. Whiteson. Softdice for imitation learning: Rethinking off-policy distribution matching. *arXiv*, 2106.03155, 2021.

G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *Proceeding of the 38th International Conference on Machine Learning*, pages 10022–10032, 2021.

U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems 20*, pages 1449–1456, 2007.

U. Syed and R. E. Schapire. A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems 23*, pages 2253–2261, 2010.

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, Ç. Gülçehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019.

References vi

T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba. Benchmarking model-based reinforcement learning. *arXiv*, 1907.02057, 2019.

Y. Wang, T. Liu, Z. Yang, X. Li, Z. Wang, and T. Zhao. On computation and generalization of generative adversarial imitation learning. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

T. Xu, Z. Li, and Y. Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems 33*, pages 15737–15749, 2020.

T. Xu, Z. Li, Y. Yu, and Z.-Q. Luo. On generalization of adversarial imitation learning and beyond. *arXiv*, 2106.10424, 2022.

Y. Zhang, Q. Cai, Z. Yang, and Z. Wang. Generative adversarial imitation learning with neural network parameterization: Global optimality and convergence rate. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11044–11054, 2020.