Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 6: Imitation Learning

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-568 (Spring 2024)

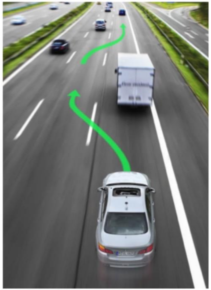
License Information for Reinforcement Learning (EE-568)

Learning from Demonstration (LfD)

Motivation:

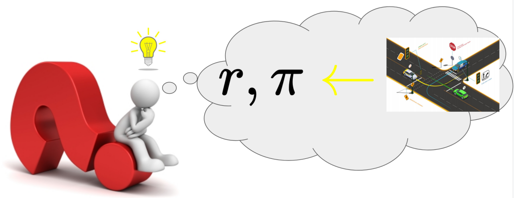
- ◦ In RL, the reward function is known and we maximize the cumulative reward.
- ◦ The reward functions are often manually designed to define the task.
- ◦ Can we instead learn a policy by capitalizing an expert's behavior?

(a)



(b)

Learning from demonstration (LfD) (cont'd)



Real world problems:

- The reward function is unknown or is difficult to be designed.
- It is easier/more natural to use "demonstrations" by experts.

Imitation learning (IL) vs inverse reinforcement learning (IRL)

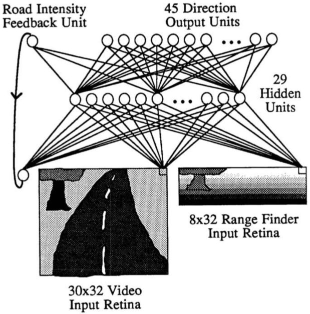
- Setting:
  - Given an expert's demonstrations $\{(s_i, \pi_{\mathsf{E}}(s_i))\}$ (offline trajectories or online queries)
  - Reward signal is unobserved
  - Transition model may be known or unknown

- Goals and approaches:
  - Recover the expert's policy π_{E} directly: imitation learning (IL)
  - Recover the expert's latent reward function $r_{\mathsf{true}}(s, a)$: inverse reinforcement learning (IRL)

A historic application

◦ Inverse Reinforcement Learning has been formally introduced by [28].

(c)



(d)

Figure: One of the first imitation learning systems using neural networks.

◦ ALVINN: Autonomous Land Vehicle In a Neural Network, 1989 [31].
https://www.youtube.com/watch?v=2KMAAmkz9go&t=205s.

One of the latest applications

- Large language models: ChatGPT

https://www.forbes.com/sites/bernardmarr/2022/12/28/what-does-chatgpt-really-mean-for-businesses/?sh=27bc344f7d1e

- The last training step is based on Reinforcement Learning from Human Feedback (RLHF) (see [29]).
- A recent work [41] shows a close connection between IRL and RLHF.

More applications

- Simulated highway driving [2]
- Helicopter acrobatics [1]
- Urban navigation [42]
- Human goal inference [24]
- Object manipulation [37, 13]

(a)

(b)

Figure: Helicopter model and instance of its acrobatics [11].

Big Picture: Taxonomy of learning from demonstration methods

Method	Reward learning	Access to environment	Interactive demonstrations	Pre-collected demonstrations
Behavioural Cloning	NO	NO	NO	YES
Online IL	NO	YES	YES	MAYBE
Inverse RL	YES	YES	NO	YES
Adversarial IL	MAYBE	YES	NO	YES
Non-adversarial IL	MAYBE	YES	NO	YES

Remarks:
- BC avoids interaction with the environment, but can suffer from cascading errors.
- Online IL helps with the cascading errors but requires (expensive) expert queries.
- IRL explains the expert's behavior but has poor sample complexity and scalability.
- Adversarial IL avoids solving RL repeatedly but is unstable due to adversarial training.
- Non-adversarial IL enjoys stable performance but has limited theoretical understanding.

Offline imitation learning: Behavioral cloning

- We assume there is an expert that has the optimal policy π_{E}.
- Input: offline data from expert's demonstration $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^{n}$, where $a_i \sim \pi_{\mathsf{E}}(s_i)$.

Offline imitation learning: Behavioral cloning

- We assume there is an expert that has the optimal policy π_{E}.
- Input: offline data from expert's demonstration $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^n$, where $a_i \sim \pi_{\mathsf{E}}(s_i)$.
- **Idea:** Directly learn the expert's policy via supervised learning.

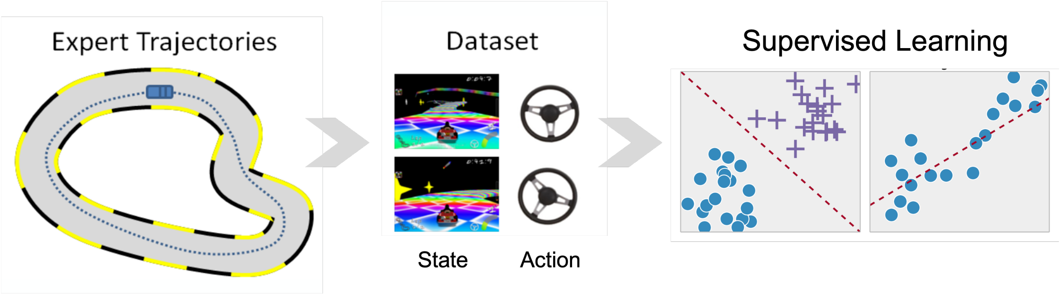


Figure: Source: `https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c`

Behavioral cloning

Maximum Likelihood Estimation (MLE)

The maximum likelihood estimator for the policy can be written as follows:

$$\hat{\pi}_{\mathsf{MLE}} = \mathrm{argmax}_{\pi \in \Pi} \sum_{(s,a) \in \mathcal{D}} \log \pi(a|s). \tag{1}$$

Risk Minimization [4]

Alternatively, we can try to minimize a loss between our parameterized policy π_θ and the expert policy π_{E} as

$$\min_{\theta} \mathbb{E}_{s \sim \lambda_\mu^{\pi_{\mathsf{E}}}(\cdot|s)} \left[\ell\Big(\pi_\theta(\cdot|s), \pi_{\mathsf{E}}(\cdot|s)\Big) \right], \tag{2}$$

where $\lambda_\mu^{\pi_{\mathsf{E}}}$ is the state visitation distribution under policy π_{E} and ℓ is a loss function. Typically, the loss function is the relative entropy.

Theoretical guarantees of BC

Theorem (Behavior Cloning) [4]

Let Π be a discrete and realizable policy class, i.e., $\pi_{\mathsf{E}} \in \Pi$. With probability at least $1-\delta$, the MLE behavioral cloning returns a policy that obeys the following guarantee on the reward J:

$$\underbrace{\langle \mu, V^{\pi_{\mathsf{E}}} \rangle}_{J(\pi_{\mathsf{E}})} - \underbrace{\langle \mu, V^{\hat{\pi}_{\mathsf{MLE}}} \rangle}_{J(\hat{\pi}_{\mathsf{MLE}})} = \langle \mu, V^{\pi_{\mathsf{E}}} - V^{\hat{\pi}_{\mathsf{MLE}}} \rangle \leq \mathcal{O}\left(\frac{1}{(1-\gamma)^2} \sqrt{\frac{\log\left(|\Pi| / \delta\right)}{|\mathcal{D}|}} \right),$$

where $|\Pi|$ is the size of the policy class, and $|\mathcal{D}|$ is the length of the provided dataset.

Theoretical guarantees of BC

Theorem (Behavior Cloning) [4]

Let Π be a discrete and realizable policy class, i.e., $\pi_{\mathsf{E}} \in \Pi$. With probability at least $1-\delta$, the MLE behavioral cloning returns a policy that obeys the following guarantee on the reward J:

$$\underbrace{\langle \mu, V^{\pi_{\mathsf{E}}} \rangle}_{J(\pi_{\mathsf{E}})} - \underbrace{\langle \mu, V^{\hat{\pi}_{\mathsf{MLE}}} \rangle}_{J(\hat{\pi}_{\mathsf{MLE}})} = \langle \mu, V^{\pi_{\mathsf{E}}} - V^{\hat{\pi}_{\mathsf{MLE}}} \rangle \leq \mathcal{O}\left(\frac{1}{(1-\gamma)^2} \sqrt{\frac{\log\left(|\Pi| / \delta\right)}{|\mathcal{D}|}} \right),$$

where $|\Pi|$ is the size of the policy class, and $|\mathcal{D}|$ is the length of the provided dataset.

Remarks:

- BC only ensures the learned policy $\hat{\pi}_{\mathsf{MLE}}$ is close to π_{E} under the support of distribution $\lambda_\mu^{\pi_{\mathsf{E}}}$.
- The term $\sqrt{\frac{\log(|\Pi|/\delta)}{|\mathcal{D}|}}$ reflects the error $\hat{\pi}_{\mathsf{MLE}}$ and π_{E} under the distribution $\lambda_\mu^{\pi_{\mathsf{E}}}$.
- The term $\frac{1}{(1-\gamma)^2}$ reflects the cascading errors when performing with respect to the policy $\hat{\pi}_{\mathsf{MLE}}$.
- The quadratic dependency on the effect horizon $H = \frac{1}{1-\gamma}$ is not avoidable in the worst case [35].
- The term $\frac{1}{(1-\gamma)^2}$ can be improved to $\frac{1}{1-\gamma}$ when the transition model is known [4].

Behavioral cloning: Advantages and disadvantages

- Advantages
  - Simple.
  - Effective. For example in ALVINN [31].

Behavioral cloning: Advantages and disadvantages

- Advantages
  - Simple.
  - Effective. For example in ALVINN [31].

- Disadvantages
  - No long-term planning.
  - Cascading errors.
  - Possible mismatch between training and testing distributions.

Quote from Pomerleau [35]

When driving for itself, the network (ALVINN) may occasionally stray from the center of road and so must be prepared to recover by steering the vehicle back to the center of the road.

A key difference with supervised learning

- The dataset $\mathcal{D}$ is collected according to π_{E}, therefore behavioural cloning outputs the policy with parameters

$$\arg\min_{\theta} \mathbb{E}_{s\sim\lambda_{\mu}^{\pi_{\mathsf{E}}}}\left[\ell\Big(\pi_\theta(\cdot|s), \pi_{\mathsf{E}}(\cdot|s)\Big)\right].$$

- However when we act in the environment with π_θ the states are sampled accordingly to λ^{π_θ}.
- Hence, ideally we would like to minimize

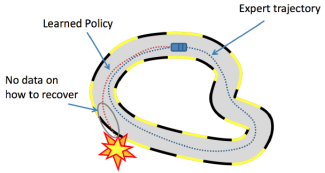
$$\min_{\theta} \mathbb{E}_{s\sim\lambda_{\mu}^{\pi_\theta}}\left[\ell\Big(\pi_\theta(\cdot|s), \pi_{\mathsf{E}}(\cdot|s)\Big)\right].$$

- Scenario different from supervised learning where the classification decisions do not affect the data distribution.

Another variation along the theme: Behavioral cloning and interactive IL

- Behavioral cloning (BC) is a supervised learning approach to learning from demonstrations
  - Given an expert's demonstrations $\{(s_i, \pi_{\mathsf{E}}(s_i))\}$ (offline trajectories or online queries)
  - Fix a loss: $\mathcal{L} : \mathcal{A} \to \mathbb{R}$
  - Output $\pi^\star \in \operatorname{argmin}_\pi \sum_i^N \mathcal{L}(a_i, \pi(s_i))$ with a_i, s_i in the dataset provided by the expert.

- BC can result in cascading errors
  - Any error at a state can accumulate over an episode.
  - It can have catastrophic consequences...



- Solution: *Interactive IL* allows to query the expert policy from a particular state

Figure: `https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c`

Interactive imitation learning

- Aims to mitigate the cascading errors through interacting with the expert.
- We assume that we can query the expert π_{E} at any time and any state sampled from $\lambda_{\mu}^{\pi_\theta}$.
- **Idea:** Learn the expert's policy via online learning.

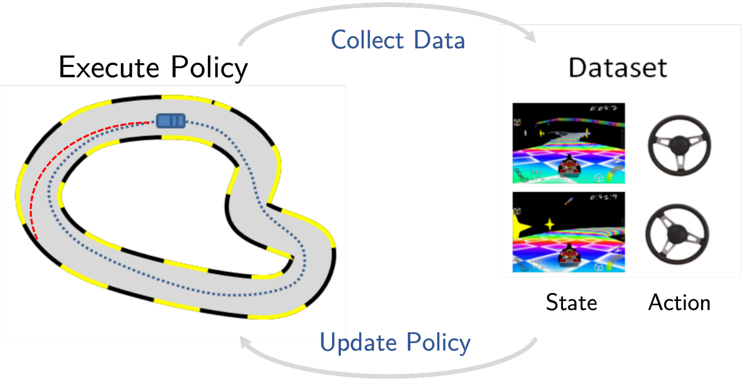


Figure: `https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c`

Interactive imitation learning

- Dataset Aggregation (DAgger) [34]: iteratively build up a policy via supervised learning on aggregated data.
- Policy Aggregation (e.g., SMILe [35]): iteratively build up a policy by mixing newly trained policies.

Interactive imitation learning

- Dataset Aggregation (DAgger) [34]: iteratively build up a policy via supervised learning on aggregated data.
- Policy Aggregation (e.g., SMILe [35]): iteratively build up a policy by mixing newly trained policies.

Interactive imitation learning

Initialize π_0

for each iteration $t = 1, \ldots, T$ **do**

  Generate trajectories τ following π_t

  Collect new data $\mathcal{D}_t = \{(s, \pi_{\mathsf{E}}(s)) | s \in \tau\}$ based on expert's feedback

  Data Aggregation: run behavioral cloning with $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_t$ and obtain π_t

  Policy Aggregation: run behavioral cloning with $\mathcal{D}_t$ and obtain $\hat{\pi}_t$, set $\pi_t = \beta\hat{\pi}_t + (1-\beta)\pi_{t-1}$

end for

Remark:
- In the dataset $\mathcal{D}_t$ the states are sampled according to λ^{π_t}.
- However, the actions are sampled from π_{E}. We need to assume that the expert is interactive.

Reduction to no-regret online learning

- Classical online optimization framework [43, 16, 10].
- Repeated game between the learner/player and the environment/adversary for any round $t = 1, \ldots, T$.

Online learning protocol

- The learner picks a decision $\mathbf{x}_t \in X$;
- The adversary picks a loss $\ell_t(\cdot) : X \to \mathbb{R}$
- The learner suffers from the loss $\ell_t(\mathbf{x}_t)$ and observes some information about ℓ_t

- The goal is to minimize the player's regret against the best decision in hindsight:

$$\mathcal{R}_T := \sum_{t=1}^{\top} \ell_t(\mathbf{x}_t) - \min_{\mathbf{x}\in X} \sum_{t=1}^{\top} \ell_t(\mathbf{x}).$$

- Follow-the-Leader Algorithm (FTL) [3]:

$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in X} \sum_{i=1}^{\top} \ell_i(\mathbf{x}), t = 1, \ldots, T$$

The reduction

$$
\begin{aligned}
\sum_{t=1}^{T} \langle \mu, V^{\pi_\mathsf{E}} - V^{\pi_t} \rangle &= \sum_{t=1}^{T} \frac{1}{1-\gamma} \mathbb{E}_{s \sim \lambda_\mu^{\pi_t}} [\langle Q^{\pi_\mathsf{E}}(s, \cdot), \pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s) \rangle] \qquad \text{(PDL)} \\
&\leq \frac{\max_{s,a} |Q^{\pi_\mathsf{E}}(s,a)|}{1-\gamma} \sum_{t=1}^{T} \mathbb{E}_{s \sim \lambda_\mu^{\pi_t}} [\|\pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s)\|_1] \\
&= \frac{\max_{s,a} |Q^{\pi_\mathsf{E}}(s,a)|}{1-\gamma} \sum_{t=1}^{T} \left(\mathbb{E}_{s \sim \lambda_\mu^{\pi_t}} [\|\pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s)\|_1] - \sum_{s \in \mathcal{D}_t} [\|\pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s)\|_1] \right) \\
&\quad + \frac{\max_{s,a} |Q^{\pi_\mathsf{E}}(s,a)|}{1-\gamma} \sum_{t=1}^{T} \sum_{s \in \mathcal{D}_t} [\|\pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s)\|_1] \\
&= \frac{\max_{s,a} |Q^{\pi_\mathsf{E}}(s,a)|}{1-\gamma} \left(\mathcal{O}(\sqrt{T}) + \mathcal{R}(T) \right)
\end{aligned}
$$

- The last inequality follows from the regret definition with losses $\ell_t(\pi) = \sum_{s \in \mathcal{D}_t} [\|\pi_\mathsf{E}(\cdot|s) - \pi_t(\cdot|s)\|]$.
- Dagger controls the regret via FTL, Smile uses an online version of conditional gradient. [16]
- The $\mathcal{O}(\sqrt{T})$ follows from Azuma-Hoeffding inequality.

Optimization perspective: DAgger

◦ DAgger is equivalent to Follow-the-Leader, which ensures no regret $o(T)$ for strongly convex loss [38].

Optimization perspective on DAgger

Let $\ell_t(\pi, \mathcal{D}_t)$ denote the behavioral cloning loss on data $\mathcal{D}_t$. At round t, DAgger minimizes the loss

$$\pi_t = \arg\min_{\pi \in \Delta} \sum_{i=1}^{T} \ell_i(\pi, \mathcal{D}_i).$$

◦ DAgger improves the error inflation factor from $\mathcal{O}\left(\frac{1}{(1-\gamma)^2}\right)$ to $\mathcal{O}\left(\frac{\max_{s,a}|Q^{\pi_{\mathsf{E}}}(s,a)|}{1-\gamma}\right)$ [4].

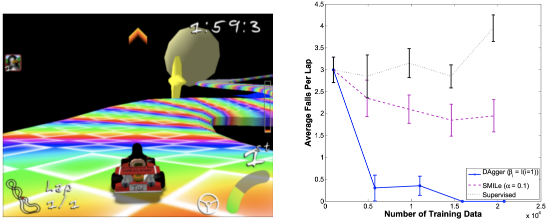


Figure: 3D racing car [34]

Feature expectation matching

- Given some features $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we define the feature expectation for π as $\rho_\phi(\pi) := \mathbb{E}_{(s,a)\sim\lambda_\mu^\pi}[\phi(s,a)]$.
- Note that $\left\|\rho_\phi(\pi_\mathsf{E}) - \rho_\phi(\pi)\right\|_2$ upper bounds the suboptimality of the policy π.

$$\langle \mu, V^{\pi_\mathsf{E}} - V^\pi \rangle \leq \frac{1}{1-\gamma}(w^\top \rho_\phi(\pi_\mathsf{E}) - w^\top \rho_\phi(\pi)) \;\leq\; \frac{1}{1-\gamma}\|w\|_2 \left\|\rho_\phi(\pi) - \rho_\phi(\pi_\mathsf{E})\right\|_2 .$$

- Therefore, solving the following problem suffices to obtain an error inflated at most by $(1-\gamma)^{-1}$:

$$\min_\pi \left\|\rho_\phi(\pi) - \rho_\phi(\pi_\mathsf{E})\right\|_2^2 . \tag{3}$$

Apprenticeship learning formalism

Assume that $r_{\mathsf{true}} \in \mathcal{R}$. Apprenticeship learning can be captured by the following problem template:

$$\min_\pi \max_{r\in\mathcal{R}} \quad J_r(\pi_\mathsf{E}) - J_r(\pi) = \min_\pi \max_{r\in\mathcal{R}} \langle \lambda_\mu^{\pi_\mathsf{E}} - \lambda_\mu^\pi , r\rangle. \tag{4}$$

Remark:
- When $\mathcal{R} = \{\sum_{i=1}^d w_i\phi_i \mid \|w\|_2 \leq 1\}$ the minimax problem (4) is reduced to (3).
- $\max_{r\in\mathcal{R}}\langle \lambda_\mu^{\pi_\mathsf{E}} - \lambda_\mu^\pi , r\rangle$ is a distance and is an integral probability metric [27] between λ_μ^π and $\lambda_\mu^{\pi_\mathsf{E}}$.
- Different choices of $\mathcal{R}$ lead to different $\mathcal{R}$-distances.

Maximum entropy inverse reinforcement learning [Ziebart et al, 2008 [42]]

◦ Consider the constrained optimization for feature expectation matching:

Max-Ent IRL

Let λ_μ^π be the state-action occupancy measure of policy π. Consider the following problem:

$$\min_{w} \max_{\pi \in \Pi} \quad w^\top \left(\mathbb{E}_{(s,a)\sim\lambda_\mu^\pi}[\phi(s,a)] - \mathbb{E}_{(s,a)\sim\lambda_\mu^{\pi_\mathsf{E}}}[\phi(s,a)] \right) + \alpha \mathbb{E}_{(s,a)\sim\lambda_\mu^\pi}[-\log \pi(a|s)].$$

Remark: ◦ Game-theoretic perspective: zero-sum game between the reward and the policy.

◦ Adding a strongly convex term in the primal is a technique known as "smoothing" in optimization.

Solving the saddle point problem

- Let $f(w) = \max_{\pi\in\Pi} \quad w^\top \left(\mathbb{E}_{s,a\sim\lambda_\mu^\pi}[\phi(s,a)] - \mathbb{E}_{s,a\sim\lambda_\mu^{\pi_\mathsf{E}}}[\phi(s,a)]\right) + \alpha\mathbb{E}_{s,a\sim\lambda_\mu^\pi}[-\log\pi(a|s)]$.
- Evaluating $f(w)$ requires solving an RL problem with reward $w^\top\phi(s,a) - \alpha\log\pi(a|s)$.
- Let $\pi^\star$ be the optimal policy for this reward.
- By Danskin's theorem [12], we can compute $\nabla_w f(w) = \left(\mathbb{E}_{s,a\sim\lambda_\mu^{\pi^\star}}[\phi(s,a)] - \mathbb{E}_{s,a\sim\lambda_\mu^{\pi_\mathsf{E}}}[\phi(s,a)]\right)$.
- And update the reward weights w by gradient descent.

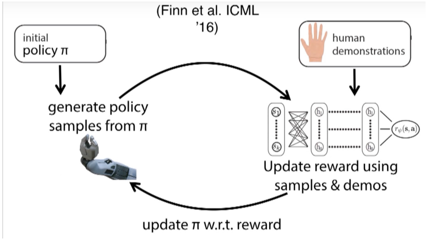
Remarks: ○ The RL step in the inner loop is expensive and it requires knowledge of the transition.

Maximum entropy inverse reinforcement learning

Max-Ent IRL Algorithm

Alternatively update

- update w by GD (with fixed π);
- update π by any RL algorithm for the corresponding entropy-regularized MDP (with fixed w)



Generative adversarial imitation learning (GAIL): A primal dual perspective

◦ In Maximum Causal Entropy IRL [42], we need to solve an RL problem for every reward update.

◦ This is a major computation bottleneck.

◦ We can develop a more efficient method if we use alternating updates.

Derivation: ◦ We will follow the same steps from [17]

GAIL objective

Let $h : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}$ be a convex function that serves as reward regularizer. GAIL solves the following minimax problem:

$$\min_{r} \max_{\pi \in \Pi} \quad \beta h(r) + \mathbb{E}_{s,a\sim\lambda_{\mu}^{\pi}}[r(s,a)] - \mathbb{E}_{s,a\sim\lambda_{\mu}^{\pi_{\mathsf{E}}}}[r(s,a)] + \alpha \mathbb{E}_{s,a\sim\lambda_{\mu}^{\pi}}[-\log \pi(a|s)]$$

◦ Use Fenchel conjugation, we can obtain

$$\max_{\pi \in \Pi} \quad -h^*(\lambda_{\mu}^{\pi_{\mathsf{E}}} - \lambda_{\mu}^{\pi}) + \alpha \mathbb{E}_{s,a\sim\lambda_{\mu}^{\pi}}[-\log \pi(a|s)].$$

◦ Important result: If f is α strongly convex then the convex conjugate $f^\star$ is $1/\alpha$-smooth [6].

An important choice for the regularizer h.

- Choosing h as

$$h(r) = \begin{cases} \mathbb{E}_{s,a\sim\lambda_\mu^{\pi_\mathsf{E}}}\left[g(r(s,a))\right], & \text{if } r(s,a) < 0; \\ \infty, & \text{otherwise.} \end{cases}$$

  with $g(x) = -x - \log(1-e^x)$.

- The Fenchel conjugate of h is given by:

$$h^*(\lambda_\mu^{\pi_\mathsf{E}} - \lambda_\mu^{\pi}) = \max_{D\in[0,1]} \mathbb{E}_{s,a\sim\lambda_\mu^{\pi}}\left[\log D(s,a)\right] + \mathbb{E}_{s,a\sim\lambda_\mu^{\pi_\mathsf{E}}}\left[\log(1-D(s,a))\right]$$

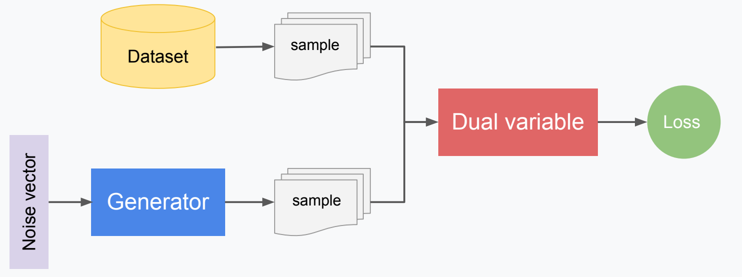
  that is widely known as the (vanilla) GAN loss.

- Therefore, we can learn a policy from demonstrations solving the following saddle point problem:

$$\min_{\pi\in\Pi}\max_{D\in[0,1]} \mathbb{E}_{s,a\sim\lambda_\mu^{\pi}}\left[\log D(s,a)\right] + \mathbb{E}_{s,a\sim\lambda_\mu^{\pi_\mathsf{E}}}\left[\log(1-D(s,a))\right] - \alpha\mathbb{E}_{s,a\sim\lambda_\mu^{\pi}}[-\log\pi(a|s)].$$

Generative Adversarial Network (GANs)

- GAN [15] is framed as a min-max game between a generator and a discriminator.



- GAN: ($\Rightarrow$ minimizing the Jensen-Shannon divergence)

$$\min_{G_\phi} \max_{D_\theta} \mathbb{E}_{x \sim p_{\text{data}}} [\log D_\theta(x)] + \mathbb{E}_z \left[\log(1 - D_\theta(G_\phi(z)))\right]$$

- Wasserstein GAN: ($\Rightarrow$ minimizing the Wasserstein divergence)

$$\min_{G_\phi} \max_{f_\theta : \text{1-Lipschitz}} \mathbb{E}_{x \sim p_{\text{data}}} [f_\theta(x)] - \mathbb{E}_z \left[f_\theta(G_\phi(z))\right]$$

Generative Adversarial Networks (GANs)

2014
GAN

2018
GAN

Generative Adversarial Imitation Learning (GAIL)

◦ GAIL [18] aims to solve the min-max game for learning the policy given an expert policy π_{E}.

$$\min_{\theta} \max_{\boldsymbol{\phi}} \quad \mathbb{E}_{s,a\sim\lambda^{\pi_\theta}}\left[\log(D_{\boldsymbol{\phi}}(s,a))\right] + \mathbb{E}_{s,a\sim\lambda_{\mu}^{\pi_{\mathsf{E}}}}\left[\log(1-D_{\boldsymbol{\phi}}(s,a))\right] - \alpha H(\pi_\theta).$$

Remarks:

◦ We assume a differentiable parametrized policy π_θ.

◦ The discriminator tries to separate the data generated from learned policy from expert data.

◦ Equivalent to minimize the Jensen-Shannon divergence between the state-action distributions of the expert policy and the learned policy.

◦ Unlike Max-Entropy IRL, does not require expensive RL subrountines to learn the reward.

Numerical performance [18]

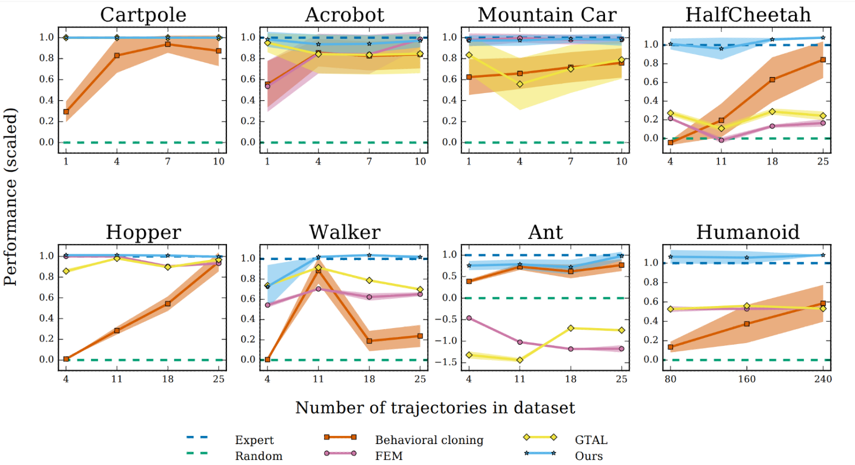


Figure: Performance of learned policies among GAIL, Behavior Cloning (BC), Feature Expectation Matching (FEM), and Game-theoretic Apprenticeship Learning (GATL)

Linear programming approach for imitation learning

- Let $\mathcal{R}$ be a class of reward functions.
- The following LP outputs the occupancy measure under the worst case reward in $\mathcal{R}$.

LP for imitation learning

$$\max_{\lambda} \min_{r \in \mathcal{R}} \langle \lambda - \lambda_{\mu}^{\pi_{\mathsf{E}}}, r \rangle \tag{5}$$

$$\text{s.t.} \quad E^{\top}\lambda = \gamma P^{\top}\lambda + (1-\gamma)\mu \tag{6}$$

Remarks:

- There are $|\mathcal{S}| + |\mathcal{S}||\mathcal{A}|$ decisions variables.
- There are $|\mathcal{S}|$ constraints.
- To avoid the large number of constraints, [23] propose to study the Lagrangian.
- To reduce the number of decision variables, [23] uses linear function approximation.

The Lagrangian

- Let $\mathcal{R}$ be a class of reward functions such that $r_{\text{true}} \in \mathcal{R}$
- The following LP outputs the occupancy measure under the worst case reward in $\mathcal{R}$.

Saddle point formulation for imitation learning

$$\max_{\lambda} \min_{r \in \mathcal{R}} \min_{V} \langle \lambda - \lambda_{\mu}^{\pi_{\mathsf{E}}}, r \rangle + \langle V, -E^{\top}\lambda + \gamma P^{\top}\lambda + (1-\gamma)\mu \rangle \tag{7}$$

Remarks:

- Notice that the number of decision variables is $|\mathcal{S}| + 2|\mathcal{S}||\mathcal{A}|$.
- Hence, we can parameterize the occupancy measure as $\lambda_{\theta} = \Phi\theta$, $V_w = \Psi w$ and $r = C\beta$.
- This parametrization helps reduce the number of decision variables significantly.
- The value parametrization has precedence in earlier RL literature.
- The occupancy measure parameterization is done out of necessity.

The reduced Lagrangian

- Introducing the linear function approximation we obtain the reduced Lagrangian.
- The number of decision variables is now $\dim(\theta) + \dim(w) + \dim(\beta)$.

Saddle Point for imitation learning

$$\max_{\theta \in \Delta} \min_{\beta \in \Delta} \min_{\|w\|_\infty \leq C} \langle \Phi\theta - \lambda_\mu^{\pi_{\mathsf{E}}}, C\beta \rangle + \langle \Psi w, -E^\top \Phi\theta + \gamma P^\top \Phi\theta + (1-\gamma)\mu \rangle \tag{8}$$

Remarks:

- We can solve the problem applying stochastic mirror prox [21].
- With this approach we get an ϵ optimal policy with $\mathcal{O}(\epsilon^{-2})$ samples.
- The sample complexity is independent of $|\mathcal{S}|$ and $|\mathcal{A}|$ due to the parametrization.
- A drawback is that one needs a strong assumption on the feature choice (see [23, 7]).

The Linear MDP Assumption

Linear MDP [20]

There exist mappings $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ and $g : \mathcal{S} \to \mathbb{R}^m$ and a vector $w \in \mathcal{W} := \{w \in \mathbb{R}^m : \|w\|_2 \leq 1\}$ such that

$$r(s,a) = \langle \phi(s,a), w \rangle$$
$$P(s'|s,a) = \left\langle \phi(s,a), g(s') \right\rangle$$

that is, in matrix form

$$r = \Phi w$$
$$P = \Phi M$$

Remarks:

- The Linear MDP is a standard setting in RL theory literature.
- It justifies an alternative LP formulation.

The constraint splitting trick

- $\mathtt{P}^2\mathtt{IL}$ [40] is derived from the primal problem for imitation learning.
- We plug in the (Linear MDP) structure in (Primal IL) (5) and we split the as follows [1]

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \min_{w \in \mathcal{W}} \langle \lambda - \lambda_{\pi_\mathsf{E}}, \Phi w \rangle$$
$$\text{s.t.} \quad E^\intercal \lambda = (1-\gamma)\mu + \gamma M^\intercal \Phi^\intercal \lambda$$

$$\Downarrow$$

$$\max_{\rho \in \Delta^m, \lambda \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \min_{w \in \mathcal{W}} \left\langle \rho - \Phi^T \lambda_\mu^{\pi_\mathsf{E}}, w \right\rangle$$
$$\text{s.t.} \quad E^\top \lambda - \gamma M^\top \rho = (1-\gamma)\mu$$
$$\Phi^\top \lambda = \rho$$

- Now we can apply on the Lagrangian, inexact proximal point updates for λ and ρ.

[1] A similar trick appeared outside the imitation learning in [26], [25] and [8]

The algorithm: $\mathtt{P^2IL}$

Proximal Point Imitation Learning: $\mathtt{P^2IL}$

Initialize π_0 as uniform distribution over $\mathcal{A}$

for $k = 1, \dots K$ **do**

`// Policy evaluation`

$$(w_k, \theta_k) \approx \operatorname*{arg\,min}_{w \in \mathcal{W}, \theta \in \Theta} \mathcal{G}_k(w, \theta)$$

`// Policy improvement`

$$\pi_k(a|s) \propto \pi_{k-1}(a|s)\, e^{-\alpha Q_{\boldsymbol{\theta}_k}(s,a)}$$

end for

- $\mathcal{G}_k(w, \theta)$, called logistic Bellman error [8], is the following convex and smooth function:

$$\mathcal{G}_k(w, \theta) \triangleq \frac{1}{\eta} \log \sum_{i=1}^{m} (\Phi^\top \lambda_{k-1})(i) e^{\eta \delta^k_{w,\theta}(i)} + (1-\gamma) \left\langle \mu, V^k_{\boldsymbol{\theta}} \right\rangle - \left\langle \lambda_{\pi_{\mathsf{E}}}, \Phi^\top w \right\rangle,$$

$$\delta^k_{w,\theta} \triangleq w + \gamma M V^k_\theta - \theta \quad \text{and} \quad V^k_\theta \triangleq \frac{1}{\alpha} \log \left(\sum_a \pi_{\lambda_{k-1}}(a|s) e^{\alpha Q_\theta(s,a)} \right) \quad \text{where} \quad Q_\theta = \Phi\theta$$

Sample Complexity Guarantees for $\mathtt{P}^2\mathtt{IL}$

- We consider errors in the maximization of $\mathcal{G}_k(w, \boldsymbol{\theta})$, i.e. $\epsilon_k = \mathcal{G}_k(w_k^\star, \theta_k^\star) - \mathcal{G}_k(w_k, \theta_k)$.
- First, we show how errors propagate.
- Second, we control that the errors are small using a Biased Stochastic Gradient Ascent subroutine.

Error propagation

Let $\widehat{\pi}_K$ be the average iterate. Then, with probability at least $1-\delta$, it holds that

$$d_{\mathcal{C}}(\lambda_{\hat{\pi}_K}, \lambda_{\pi_{\mathrm{E}}}) \leq \frac{1}{K}\left(\log\left(m\,|\mathcal{A}|\right) + C\sum_k \sqrt{\epsilon_k} + \sum_k \epsilon_k\right).$$

Error control

Let (w_k, θ_k) be the output of the Biased Stochastic Gradient Ascent subroutine for T iterations. Then, $\epsilon_k = \max_{w,\theta} \mathcal{G}_k(w, \theta) - \mathcal{G}_k(w_k, \theta_k) \leq \mathcal{O}(\frac{\max\{\eta,1\}m}{\beta\sqrt{T}})$, with probability $1-\delta$.

A downside: exploration assumptions

Remarks:

- Choosing $K = \Omega(\epsilon^{-1})$ and $T = \Omega(\epsilon^{-4})$ we obtain $\mathcal{O}(\epsilon^{-5})$ sample complexity.
- We use samples to approximate the gradients $\nabla_\theta \mathcal{G}_k$ and $\nabla_w \mathcal{G}_k$.
- In REPS, [30] required the following assumption.

Exploration assumption

We can sample state action pairs from an occupancy measure $\lambda_{\pi_0}(s, a) > 0 \quad \forall s, a \in \mathcal{S} \times \mathcal{A}$.

- In our extension to Linear MDP, we require the following assumption.

Positive Definite Covariance Matrix

We can sample state action pairs from an occupancy measure λ_{π_0} such that.

$$\sigma_{\min}\left(\mathbb{E}_{s,a\sim\lambda_{\pi_0}} \phi(s,a)\phi(s,a)^\top\right) \geq \beta > 0.$$

Guarantees for ILARL [2]

Theorem

After using $\widetilde{\mathcal{O}}\left(\frac{\log|\mathcal{A}|d^3}{(1-\gamma)^8\epsilon^4}\right)$ *state action pairs from the MDP and using* $\widetilde{\mathcal{O}}\left(\frac{2d\log(2d)}{(1-\gamma)^2\epsilon_E^2}\right)$ *expert demonstrations ILARL outputs a policy which is at most* $\epsilon+\epsilon_E$*-suboptimal, i.e.*

$$\mathbb{E}[\langle \mu, V^{\pi^\star} - V^{\pi^{\mathrm{out}}}\rangle] \leq \epsilon + \epsilon_E$$

Remarks:

- No RL in the inner loop.
- No need to know the transitions.
- It bypasses the use of a generative model or the use of exploration assumptions.

[2] Viano, Skoulakis and Cevher "Imitation Learning in Discounted Linear MDP without exploration assumptions.", Under Review

Is imitating enough ?

- Standard imitation learning
  - copy the *actions* performed by the expert
  - no reasoning about outcomes of actions

Figure: Robot imitation

- Human imitation learning
  - copy the *intent* of the expert
  - might take very different actions!

Figure: Human imitation

Inverse reinforcement learning (IRL) [28, 36]

IRL Objective

Find reward function $r(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to [-1, 1]$ that explains the expert's behavior:

$$\pi_{\mathsf{E}} \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi\right].$$

Inverse reinforcement learning (IRL) [28, 36]

IRL Objective

Find reward function $r(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to [-1, 1]$ that explains the expert's behavior:

$$\pi_{\mathsf{E}} \in \arg\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi\right].$$

Namely, it holds that

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi_{\mathsf{E}}\right] \geq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi\right], \forall \pi \in \Pi.$$

Remarks:

- Assume the expert is optimizing some reward function r_{true}.
- The true reward function is unknown; π_{E} is the optimal policy of the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r_{\mathsf{true}}, \gamma)$.
- Unlike BC, IRL uses the MDP structure for the learning from expert demonstration.
- IRL recovers a reward function and avoids the distribution shift issue in BC [2, 42].
- Note that this is a convex feasibility problem: It has different solution challenges.

The RL and IRL dichotomy

	IRL	**RL**
Input	Expert Demonstrations	Reward Function
Output	Optimal policy Reward function	Optimal Policy

◦ RL recovers a nearly optimal behavior from reward functions.

◦ IRL recovers a reward function for which the observed behaviour is optimal and possibly a nearly optimal behavior from demonstrations by an expert.

Challenges with inverse reinforcement learning

Theorem (Reward shaping)

An expert policy π_E *optimal in the MDP* $\mathcal{M}$ *with reward* r *is optimal also in the MDP* $\mathcal{M}$ *with reward function* $\hat{r}$ *given by*

$$\hat{r}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\Phi(s') \right] - \Phi(s),$$

where $\Phi : \mathcal{S} \to \mathbb{R}$ *is called potential function.*

- Reward function ambiguity; A trivial solution is $r = 0$.
  - Solution: Add regularization, restrict reward assumptions
- IRL is computationally expensive if we want to enumerate all polices to form the constraints.
  - Solution: Consider a tractable apprenticeship learning formalism
- In practice, we do not observe π_E but only trajectories from π_E.
  - Solution: Use sample averages of total returns under π_E
- May be infeasible if the expert's policy is not optimal.
  - Solution: Relax the constraints; add slack variables

Identifiability in inverse reinforcement learning

◦ The reward function ambiguity problem can be solved leveraging two experts. The following holds:

Theorem (Theorem 2 in [33])

Consider two Markov decision problems on the same set of states and actions, but with different transition matrices P^1, P^2 and discount factors γ_1, γ_2. Suppose that we observe two experts acting each in one of these environments, optimally with respect to the same reward function, in the sense that their policies maximize the entropy regularized reward in their respective environments. Then, the reward function can be recovered up to the addition of a constant if and only if

$$\mathit{rank}\begin{pmatrix} I - \gamma_1 P^1_{a_1} & -(I - \gamma_2 P^2_{a_1}) \\ \vdots & \vdots \\ I - \gamma_1 P^1_{a_{|\mathcal{A}|}} & -(I - \gamma_2 P^2_{a_{|\mathcal{A}|}}) \end{pmatrix} = 2|\mathcal{S}| - 1. \tag{9}$$

Remark:
- ◦ This result has been stated very recently in [9] under a limited form.
- ◦ This stronger statement is a new result.
- ◦ Identifying the reward is important when one needs to predict how the expert would behave under different dynamics but same reward.

Feature-based reward

Theorem

Assumption Let $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ be a feature mapping. Assume linear true reward function, i.e.,

$$r_{\mathit{true}} \in \{r \mid r(s,a) = w^\top \phi(s,a),\ \text{where } w \in \mathbb{R}^d \text{ and } \|w\|_2 \leq 1\}.$$

◦ The expected total reward when $r(s,a) = w^\top \phi(s,a)$ can then be expressed as:

$$J_r(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \middle| \pi\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t,a_t) \middle| \pi\right] = w^\top \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t,a_t) \middle| \pi\right] = w^\top \rho_\phi(\pi),$$

where $\rho_\phi(\pi) \in \mathbb{R}^d$ is the feature expectation vector of policy π.

Goal

Find $w \in \mathbb{R}^d$ such that

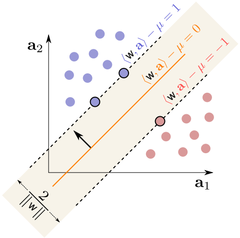
$$\underbrace{w^\top \rho_\phi(\pi_{\mathsf{E}})}_{=J_w(\pi_{\mathsf{E}})} \geq \underbrace{w^\top \rho_\phi(\pi)}_{=J_w(\pi)},\ \forall \pi \in \Pi.$$

Feature-based reward (cont'd)

Goal

Find $w \in \mathbb{R}^d$ such that

$$\underbrace{w^\top \rho_\phi(\pi_{\mathsf{E}})}_{=J_w(\pi_{\mathsf{E}})} \geq \underbrace{w^\top \rho_\phi(\pi)}_{=J_w(\pi)}, \ \forall \pi \in \Pi.$$



Remark:

- Note that $\rho_\phi(\pi)$ can be readily estimated from sampled trajectories.
- By Hoeffding's Lemma [19] (see 12) we need $\mathcal{O}\left(\frac{d \log(\frac{1}{\delta})}{(1-\gamma)^2 \varepsilon^2}\right)$ expert trajectories to have an ε-small ℓ_∞-error with probability at least $1-\delta$.

Max margin IRL [Ratliff et al., 2006][32]

Standard max-margin formulation [39]

We want to maximize the *margin*, i.e the separation distance between the expert and other policies, this yields

$$\begin{aligned} \min_w \quad & \|w\|_2^2 \\ \text{s.t.} \quad & w^\top \rho_\phi(\pi_\mathsf{E}) \ \geq \ w^\top \rho_\phi(\pi) + 1, \quad \text{for all } \pi \end{aligned}$$

Structured prediction max margin

We add flexibility by specifying the margin as a function of the policies, i.e., $m(\pi_\mathsf{E}, \pi)$, this yields

$$\begin{aligned} \min_w \quad & \|w\|_2^2 \\ \text{s.t.} \quad & w^\top \rho_\phi(\pi_\mathsf{E}) \ \geq \ w^\top \rho_\phi(\pi) + m(\pi_\mathsf{E}, \pi), \quad \text{for all } \pi \end{aligned}$$

Remarks:
- ◦ We want to make $J_w(\pi_\mathsf{E})$ larger than any other $J_w(\pi)$ by a margin $m(\pi_\mathsf{E}, \pi)$.
- ◦ Margin should be larger for policies that are very different from π_E.
- ◦ Example: $m(\pi_\mathsf{E}, \pi) =$number of states in which π_E was observed and in which π and π_E disagree.

Max margin IRL [Ratliff et al., 2006][32] (cont')

Structured prediction max-margin with slack variables

We relax the problem by allowing the constraints to be violated by introducing slack variables $\xi \geq 0$, this yields

$$\begin{aligned} \min_{w,\xi} \quad & \|w\|_2^2 + C\xi \\ \text{s.t.} \quad & w^\top \rho_\phi(\pi_{\mathsf{E}}) \geq w^\top \rho_\phi(\pi) + m(\pi_{\mathsf{E}}, \pi) - \xi, \quad \text{for all } \pi \end{aligned}$$

Remarks:

- The slack variable $\xi \geq 0$ are introduced to allow the constraints to be violated.
- Resolved: access to π_{E}, reward ambiguity, expert suboptimality.
- One challenge remains: very large number of constraints.
- Assuming access to an RL subroutine, it can be solved, e.g., by constraint generation.

Summary of imitation learning

Method	Reward learning	Access to environment	Interactive demonstrations	Pre-collected demonstrations
Behavioural Cloning	NO	NO	NO	YES
Online IL	NO	YES	YES	MAYBE
Inverse RL	YES	YES	NO	YES
Adversarial IL	MAYBE	YES	NO	YES
Non-adversarial IL	MAYBE	YES	NO	YES

Remarks:

- BC avoids interaction with the environment, but can suffer from cascading errors.
- Online IL helps with the cascading errors but requires (expensive) expert queries.
- IRL explains the expert's behavior but has poor sample complexity and scalability.
- Adversarial IL avoids solving the RL problem repeatedly but are unstable due to adversarial training.
- Non-adversarial IL enjoys stable performance but is hampered by limited theoretical understanding.

References I

[1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Ng.
An application of reinforcement learning to aerobatic helicopter flight.
Advances in neural information processing systems, 19, 2006.
8

[2] Pieter Abbeel and Andrew Y Ng.
Apprenticeship learning via inverse reinforcement learning.
In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.
8, 45, 46

[3] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin.
Competing in the dark: An efficient algorithm for bandit linear optimization.
In *In Proceedings of the 21st Annual Conference on Learning Theory (COLT*, 2008.
22

[4] Alekh Agarwal, Nan Jiang, Sham M. Kakade, and Wen Sun.
Reinforcement learning: Theory and algorithms, 2020.
12, 13, 14, 24

[5] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun.
Flambe: Structural complexity and representation learning of low rank mdps.
In *Neural Information Processing Systems (NeurIPS)*, 2020.
65, 66, 67

References II

[6] Jean-Bernard Baillon and Georges Haddad.
Quelques propriétés des opérateurs angle-bornés et n-cycliquement monotones.
Israel Journal of Mathematics, 26:137–150, 1977.
29

[7] J. Bas-Serrano and G. Neu.
Faster saddle-point optimization for solving large-scale Markov decision processes.
In *Conference on Learning for Dynamics and Control (L4DC)*, 2020.
37

[8] Joan Bas-Serrano, Sebastian Curi, Andreas Krause, and Gergely Neu.
Logistic Q-learning.
In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.
39, 40

[9] Haoyang Cao, Samuel N. Cohen, and Lukasz Szpruch.
Identifiability in inverse reinforcement learning, 2021.
49

[10] Nicolo Cesa-Bianchi, Gabor Lugosi, and Learning Prediction.
Games, 2006.
22

References III

[11] Adam Coates, Pieter Abbeel, and Andrew Y Ng.
Learning for control from multiple demonstrations.
In *Proceedings of the 25th international conference on Machine learning*, pages 144–151, 2008.
8

[12] J. Danskin.
The theory of max-min, with applications.
SIAM Journal on Applied Mathematics, 14(4):641–664, 1966.
27

[13] Chelsea Finn, Sergey Levine, and Pieter Abbeel.
Guided cost learning: Deep inverse optimal control via policy optimization.
In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 49–58, New York, New York, USA, 20–22 Jun 2016. PMLR.
8

[14] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon.
IQ-learn: Inverse soft-Q learning for imitation.
In *Advances in Neural Information Processing Systems*, 2021.
69

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative Adversarial Networks.
ArXiv e-prints, June 2014.
31

References IV

[16] Elad Hazan.
Introduction to online convex optimization.
Foundations and Trends® in Optimization, 2(3-4):157–325, 2016.
22, 23

[17] Jonathan Ho and Stefano Ermon.
Generative adversarial imitation learning.
In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.
29

[18] Jonathan Ho and Stefano Ermon.
Generative adversarial imitation learning.
In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
33, 34

[19] Wassily Hoeffindg.
Probability inequalities for sums of bounded random variables.
Journal of the American Statistical Association, 58(301):13–30, 1963.
51, 68

[20] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan.
Provably efficient reinforcement learning with linear function approximation.
In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.
38

References V

[21] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel.
Solving variational inequalities with stochastic mirror-prox algorithm.
Stochastic Systems, 1(1):17–58, 2011.
37

[22] Sham Kakade and John Langford.
Approximately optimal approximate reinforcement learning.
In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.
65, 66, 67

[23] Angeliki Kamoutsi, Goran Banjac, and John Lygeros.
Efficient performance bounds for primal-dual reinforcement learning from demonstrations.
In *International Conference on Machine Learning (ICML)*.
35, 37

[24] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert.
Activity forecasting.
In *European conference on computer vision*, pages 201–214. Springer, 2012.
8

[25] Donghwan Lee and Niao He.
Stochastic primal-dual q-learning algorithm for discounted mdps.
In *2019 american control conference (acc)*, pages 4897–4902. IEEE, 2019.
39

References VI

[26] Prashant G Mehta and Sean P Meyn.
Convex q-learning, part 1: Deterministic optimal control.
arXiv preprint arXiv:2008.03559, 2020.
39

[27] Alfred Müller.
Integral probability metrics and their generating classes of functions.
Advances in applied probability, 29(2):429–443, 1997.
25

[28] A. Y. Ng and S. J. Russell.
Algorithms for inverse reinforcement learning.
In *International Conference on Machine Learning (ICML)*, 2000.
6, 45, 46

[29] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.
Training language models to follow instructions with human feedback.
arXiv preprint arXiv:2203.02155, 2022.
7

[30] Aldo Pacchiano, Jonathan Lee, Peter Bartlett, and Ofir Nachum.
Near optimal policy optimization via REPS.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
42

References VII

[31] Dean A. Pomerleau.
Alvinn: An autonomous land vehicle in a neural network.
In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1989.
6, 15, 16

[32] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich.
Maximum margin planning.
In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006.
52, 53

[33] Paul Rolland, Luca Viano, Norman Schürhoff, Boris Nikolov, and Volkan Cevher.
Identifiability and generalizability from multiple experts in inverse reinforcement learning.
arXiv preprint arXiv:2209.10974, 2022.
49

[34] S. Ross, G. Gordon, and D. Bagnell.
A reduction of imitation learning and structured prediction to no-regret online learning.
In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.
20, 21, 24

[35] Stéphane Ross and Drew Bagnell.
Efficient reductions for imitation learning.
In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.
13, 14, 15, 16, 20, 21

References VIII

[36] Stuart Russell.
Learning agents for uncertain environments (extended abstract).
In *Annual Conference on Computational Learning Theory (COLT)*, 1998.
45, 46

[37] Stefan Schaal.
Is imitation learning the route to humanoid robots?, 1999.
8

[38] Shai Shalev-Shwartz.
Online learning and online convex optimization.
Foundations and Trends® in Machine Learning, 4(2):107–194, 2012.
24

[39] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin.
Learning structured prediction models: A large margin approach.
In *Proceedings of the 22nd international conference on Machine learning*, pages 896–903, 2005.
52

[40] Luca Viano, Angeliki Kamoutsi, Gergely Neu, Igor Krawczuk, and Volkan Cevher.
Proximal point imitation learning.
arXiv preprint arXiv:2209.10968, 2022.
39

References IX

[41] Banghua Zhu, Jiantao Jiao, and Michael I Jordan.
Principled reinforcement learning with human feedback from pairwise or k-wise comparisons.
arXiv preprint arXiv:2301.11270, 2023.
7

[42] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al.
Maximum entropy inverse reinforcement learning.
volume 8, pages 1433–1438. Chicago, IL, USA, 2008.
8, 26, 29, 45, 46

[43] Martin Zinkevich.
Online convex programming and generalized infinitesimal gradient ascent.
In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.
22

Supplementary Material

Proof Sketch

- Recall the advantage defined as $A^{\hat{\pi}}(s,a) = Q^{\hat{\pi}}(s,a) - V^{\hat{\pi}}(s)$ and notice that $\mathbb{E}_{a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a) = 0, \quad \forall s$.
- We will use also that $A^{\hat{\pi}}(s,a) \leq \frac{1}{1-\gamma}$ if $\max_{s,a} |r(s,a)| \leq 1$.

Proof.

- Based on performance difference lemma [22], we have

$$\begin{aligned}
V^{\pi_{\mathsf{E}}} - V^{\hat{\pi}} &= \frac{1}{1-\gamma} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) \\
&= \frac{1}{1-\gamma} \left[\mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) - \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a) \right] \\
&\leq \frac{1}{(1-\gamma)^2} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \|\hat{\pi}(\cdot|s) - \pi_{\mathsf{E}}(\cdot|s)\|_1 .
\end{aligned}$$

$\square$

Proof Sketch

- Recall the advantage defined as $A^{\hat{\pi}}(s,a) = Q^{\hat{\pi}}(s,a) - V^{\hat{\pi}}(s)$ and notice that $\mathbb{E}_{a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a) = 0, \quad \forall s.$
- We will use also that $A^{\hat{\pi}}(s,a) \leq \frac{1}{1-\gamma}$ if $\max_{s,a} |r(s,a)| \leq 1.$

Proof.

- Based on performance difference lemma [22], we have

$$
\begin{aligned}
V^{\pi_{\mathsf{E}}} - V^{\hat{\pi}} &= \frac{1}{1-\gamma} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\ a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) \\
&= \frac{1}{1-\gamma} \left[\mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\ a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) - \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\ a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a)\right] \\
&\leq \frac{1}{(1-\gamma)^2} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \|\hat{\pi}(\cdot|s) - \pi_{\mathsf{E}}(\cdot|s)\|_1 .
\end{aligned}
$$

- MLE guarantee [5] is given by

$$
\mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \|\hat{\pi} - \pi_{\mathsf{E}}\|_{TV}^2 \leq \frac{\log\left(|\Pi|/\delta\right)}{|\mathcal{D}|}.
$$

$\square$

Proof Sketch

- Recall the advantage defined as $A^{\hat{\pi}}(s,a) = Q^{\hat{\pi}}(s,a) - V^{\hat{\pi}}(s)$ and notice that $\mathbb{E}_{a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a) = 0, \quad \forall s.$
- We will use also that $A^{\hat{\pi}}(s,a) \leq \frac{1}{1-\gamma}$ if $\max_{s,a} |r(s,a)| \leq 1.$

Proof.

- Based on performance difference lemma [22], we have

$$\begin{aligned} V^{\pi_{\mathsf{E}}} - V^{\hat{\pi}} &= \frac{1}{1-\gamma} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) \\ &= \frac{1}{1-\gamma}\left[\mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\pi_{\mathsf{E}}(\cdot|s)} A^{\hat{\pi}}(s,a) - \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}},\, a\sim\hat{\pi}(\cdot|s)} A^{\hat{\pi}}(s,a)\right] \\ &\leq \frac{1}{(1-\gamma)^2} \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \|\hat{\pi}(\cdot|s) - \pi_{\mathsf{E}}(\cdot|s)\|_1. \end{aligned}$$

- MLE guarantee [5] is given by

$$\mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \|\hat{\pi} - \pi_{\mathsf{E}}\|_{TV}^2 \leq \frac{\log(|\Pi|/\delta)}{|\mathcal{D}|}.$$

- Then the result follows from Jensen's inequality and that $\|\cdot\|_{TV} = \frac{1}{2}\|\cdot\|_1$. □

$\star$ Hoeffding's Lemma [19]

Theorem (Hoeffding's Lemma)

Let X be a random variable such that $\mathbb{E}(X) = 0$ and $X \in [a, b]$ almost surely. Then for any $s \in \mathbb{R}$, it holds that

$$\mathbb{E}(e^{sX}) \leq e^{\frac{s^2(b-a)^2}{8}}.$$

The IQ-Learn optimization problem [14]

- The core idea is to use the expert data to learn a state action value function.
- We can see IQ-Learn as a double smoothing approach.
- We add a strongly convex function occupancy measure dependent function $H(\cdot|\lambda_0)$
- Analogously, we add a strongly concave function dependent on the reward variable r.

$$\min_{\lambda \in \mathfrak{F}} \max_{r} \langle \lambda_{\pi_{\mathsf{E}}} - \lambda, r \rangle + \frac{1}{\chi} \psi(r) + \frac{1}{\eta} H(\lambda, \lambda_0),$$

where H is the relative conditional entropy defined as $H(\lambda, \lambda_0) := \sum_{x,a} \lambda(x,a) \log \frac{\lambda(x,a) \sum_a \lambda_{\pi_0}(x,a)}{\lambda_{\pi_0}(x,a) \sum_a \lambda(x,a)}$.

- $\psi(r)$ is restricted to a particular form, i.e. $\psi(r) = \langle \lambda_{\pi_{\mathsf{E}}}, r - \phi(r) \rangle$, with $\phi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \to \mathbb{R}$ being a convex and non-increasing function.

IQ-Learn equivalent unconstrained problem

IQ-Learn Program over Q-functions

Replacing the optimal policy $\pi_Q(a|s) \propto \exp\left(Q(s,a)\right)$ and let $V_Q(s) = \log \sum_{a \in \mathcal{A}} \exp(Q(s,a))$, we obtain an unconstrained problem.

$$\tilde{Q} \approx \arg\max_{Q} (1-\gamma) \left\langle \mu, V_Q \right\rangle - \left\langle \lambda_{\pi_{\mathsf{E}}}, \phi\left(Q - \gamma P V_Q\right) \right\rangle$$

Remarks:

- The approach is very similar to REPS.
- However, the derivation of the unconstrained problem is not straightforward and requires assumptions on ψ.
- The formulation is concave w.r.t. Q.
- The empirical performance of this algorithm is very convincing.
- Lack of convergence guarantees.
- It solves the feature matching problem without employing minmax updates.

Beyond the exploration assumption with ILARL

◦ Algorithm obtained using ideas similar to OPPO (Check lecture 5).

Imitation Learning via Adversarial Reinforcement Learning: ILARL

1: Initialize π_0 as uniform distribution over $\mathcal{A}$
2: **for** $k = 1, \ldots K$ **do**
3: // Reward players update

$$r^{k+1} = \Pi_{\mathcal{R}}\left[r^k + \gamma(\lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi^k})\right]$$

4: // Policy players update
5: Find an estimator-uncertainty pair (θ^k, b^k) such that

$$\gamma\left|\phi(s,a)^T\theta^k - PV^k(s,a)\right| \leq b^k(s,a) \qquad \forall s, a \in \mathcal{S}\times\mathcal{A} \quad \text{with high probability.}$$

6: Update Q values

$$Q^{k+1}(s,a) = r^k(s,a) + \gamma\phi(s,a)^T\theta^k + b^k(s,a).$$

7: Update policy

$$\pi_{k+1}(a|s) \propto \pi_k(a|s)e^{\eta Q^k(s,a)}$$

8: **end for**