Advancing Infinite Horizon Imitation Learning: Efficiency Guarantees and Assumption-free Exploration

Volkan Cevher

volkan.cevher@epfl.ch

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

RL Workshop @ IISc

joint work with

Luca Viano & Angeliki Kamoutsi & Gergely Neu & Igor Krawzuck & Stratis Skoulakis

(26 February 2024)

lions@epfl swisscom E4S Enterprise for Society EPFL

Warm up

- Reinforcement learning (RL): Sequential decision making in unknown environment
- Markov decision process (MDP): $M = (\mathcal{S}, \mathcal{A}, P, r, \mu, \gamma)$
- Stationary stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi(\cdot|s_t)$
- State-value function: $V_r^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right]$
  - we drop the dependence r when the context is clear

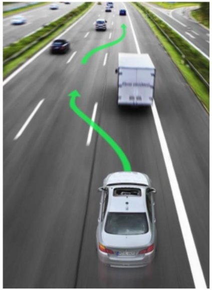
Challenges:

- Unknown dynamics: knowledge only through sampled experience.
- Large state and actions spaces.

Learning from demonstrations

- The reward function $r(a, s)$ is central to reinforcement learning pipeline
  - generally, the reward function is unknown
  - imitation of an expert may be easier than designing a reward function

(a)



(b)

Nuances: A comparison

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

- The basic setting for inverse reinforcement learning (IRL) and imitation learning (IL):
  - Given an expert's demonstrations $\mathcal{D}_{\pi_{\mathsf{E}}} = \{(s_i, a_i)\}_{i=1}^{N_{\mathsf{E}}}$
  - The true reward function r_{true} is unknown to the learner
  - Transition model is often unknown

Nuances: A comparison

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

- The basic setting for inverse reinforcement learning (IRL) and imitation learning (IL):
  - Given an expert's demonstrations $\mathcal{D}_{\pi_{\mathsf{E}}} = \{(s_i, a_i)\}_{i=1}^{N_{\mathsf{E}}}$
  - The true reward function r_{true} is unknown to the learner
  - Transition model is often unknown
- *This talk focuses on the IL setting towards the expert policy*
  - Shameless plug: our work on IRL reward identifiability with applications to finance [Rolland et al., 2022]

Solution via linear programming

- The linear programming (LP) approach
  - It formulates the RL problem as an LP.
  - Promising way to overcome the limitations of dynamic programming.

The first part of this talk is about

Provably efficient IL algorithms via **proximal point method** and the **LP** approach to MDPs.

Remark: ◦ We will discover our algorithm P^2IL [Viano et al., 2022] (NeurIPS 2022).

(Detour) Revisiting the Bellman optimality equation

- We denote $V^\star(s) = \max_{\pi\in\Pi} \; V^\pi(s)$.
- $V^\star$ satisfies the Bellman optimality equation, which can be written as a feasibility problem:

$$\begin{aligned} &\min_{V} \;\; 0 \\ &\text{s.t.} \;\; V(s) = (\mathcal{T}V)(s) := \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V(s') \right], \quad \forall\, s \in \mathcal{S}. \end{aligned}$$

  - $\mathcal{T}$ is the so-called Bellman operator
  - The only feasible assignment is $V^\star$
  - The above equality constraints are nonlinear in V due to the maximization over $\mathcal{A}$

(Detour) Revisiting the Bellman optimality equation

◦ We denote $V^\star(s) = \max_{\pi\in\Pi} V^\pi(s)$.

◦ $V^\star$ satisfies the Bellman optimality equation, which can be written as a feasibility problem:

$$\min_{V} \; 0$$
$$\text{s.t.} \;\; V(s) = (\mathcal{T}V)(s) := \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V(s') \right], \quad \forall\, s \in \mathcal{S}.$$

- $\mathcal{T}$ is the so-called Bellman operator
- The only feasible assignment is $V^\star$
- The above equality constraints are nonlinear in V due to the maximization over $\mathcal{A}$

Remarks: ◦ The Bellman optimality operator is a γ-contraction mapping w.r.t. ℓ_∞-norm:

$$\left\| \mathcal{T}V' - \mathcal{T}V \right\|_\infty \leq \gamma \left\| V' - V \right\|_\infty.$$

◦ The Bellman operator is also monotonic (component-wise): $V' \leq V \;\Rightarrow\; \mathcal{T}V' \leq \mathcal{T}V$.

(Detour) A relaxation of the Bellman optimality condition: Bellman inequalities

- The Bellman optimality $\Rightarrow V^\star$ is the function with the lowest values $V(s)$ among all $V \in \mathbb{R}^{|\mathcal{S}|}$ satisfying

$$V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V(s'), \quad \forall\, s \in \mathcal{S},\ a \in \mathcal{A}. \qquad \text{(BELLMAN INEQUALITY)}$$

- Note that the BELLMAN INEQUALITY constraint is linear in $V \implies$ Linear Programming (LP)

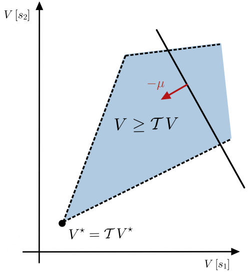


Figure: Graphical interpretation of Bellman inequality

(Detour) Solving MDPs with LP - Dual LP formulation

Dual LP

Let $\mu(s) > 0, s \in \mathcal{S}$ be the initial distribution (or any positive weights). The dual LP formulation is given by

$$
\begin{aligned}
\min_{V} \quad & (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s) V(s) \\
\text{s.t.} \quad & V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V(s'), \quad \forall\, s \in \mathcal{S},\ a \in \mathcal{A}.
\end{aligned} \tag{D}
$$

Remarks:

- The optimal value function $V^\star$ is the unique solution to the above LP.
- Number of decision variables: $|\mathcal{S}|$, number of constraints: $|\mathcal{S}| \times |\mathcal{A}|$.
- An optimal (deterministic) policy is the associated greedy policy

$$
\pi^\star(s) \in \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s) \right]. \tag{1}
$$

- The factor $(1-\gamma)$ in (D) ensures that the dual variables are in the simplex $\Delta_{\mathcal{S} \times \mathcal{A}}$.

(Detour) Solving MDPs with primal LP

Primal LP formulation

Let $\mu(s) > 0, s \in \mathcal{S}$ be the initial distribution (or any positive weights). The primal LP formulation is given by

$$\begin{aligned} \max_{\lambda \geq 0} \quad & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} r(s,a)\lambda(s,a) \\ \text{s.t.} \quad & \sum_{a \in \mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} \mathsf{P}(s|s',a')\lambda(s',a'), \quad \forall\, s \in \mathcal{S}. \end{aligned} \tag{P}$$

Remarks:

- Number of decision variables: $|\mathcal{S}| \times |\mathcal{A}|$.
- Number of constraints: $|\mathcal{S}| + |\mathcal{S}| \times |\mathcal{A}|$.
- The constraints implicitly implies the decision variables are in the probability simplex.
- The primal solution $\lambda^\star$ corresponds to the state-action occupancy measure of $\pi^\star$.

From RL to IL

Dual LP

$$\min_{V \in \mathbb{R}^{|\mathcal{S}|}} \quad (1-\gamma)\langle \mu, V \rangle \qquad \text{s.t.} \quad EV \geq r + \gamma P V. \tag{D}$$

Primal LP

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad \langle \lambda, r \rangle \qquad \text{s.t.} \quad E^\intercal \lambda = (1-\gamma)\mu + \gamma P^\intercal \lambda, \quad \lambda \geq 0. \tag{P}$$

The imitation learning goal

The goal is to learn an ϵ-optimal policy using as few resources as possible.

An ϵ-optimal policy with respect to the expert

A policy π is said ϵ-optimal policy with respect to the expert policy π_{E} if it satisfies the following:

$$(1-\gamma)\left\langle \mu, V^{\pi_{\mathsf{E}}}_{r_{\mathrm{true}}} - V^{\pi}_{r_{\mathrm{true}}} \right\rangle \leq \epsilon \quad \text{or} \quad \langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi}, r_{\mathrm{true}} \rangle \leq \epsilon.$$

Remarks: ◦ The learner tries to learn a ϵ-optimal policy using the following resources:

- **Expert demonstrations:** N_{E} state action pairs sampled from the expert.
- **Online interactions:** N state action pairs sampled from the learner occupancy measure.
- **Computation:** The number of arithmetic operation needed to approximate the expert policy.

IL via convex programming

- We can compute an estimator $\widehat{\lambda^{\pi_E}}$ of λ^{π_E} using the expert dataset $\mathcal{D}_{\pi_{\mathsf{E}}}$ as follows:

$$\widehat{\lambda^{\pi_E}}(s,a) = \frac{1}{N_E} \sum_{s',a' \in \mathcal{D}_{\pi_{\mathsf{E}}}} \mathbb{1}\{s,a = s',a'\}$$

- Assuming $r_{\text{true}} \in \mathcal{R}$, we can obtain the following useful surrogate for optimality:

$$\begin{aligned}\langle \lambda^{\pi_{\mathsf{E}}} - \lambda, r_{\text{true}} \rangle &\leq \max_{r \in \mathcal{R}} \langle \lambda^{\pi_{\mathsf{E}}} - \lambda, r \rangle = \max_{r \in \mathcal{R}} \langle \widehat{\lambda^{\pi_{\mathsf{E}}}} - \lambda, r \rangle + \left\langle \lambda^{\pi_{\mathsf{E}}} - \widehat{\lambda^{\pi_{\mathsf{E}}}}, r \right\rangle \\ &\leq \max_{r \in \mathcal{R}} \langle \widehat{\lambda^{\pi_{\mathsf{E}}}} - \lambda, r \rangle + \frac{d}{\sqrt{N_E}}.\end{aligned}$$

Primal IL formulation

While we cannot improve on the second term above, we can optimize the first term as follows:

$$\begin{aligned}\min_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad & \max_{r \in \mathcal{R}} \langle \widehat{\lambda_{\pi_{\mathsf{E}}}} - \lambda, r \rangle \\ \text{s.t.} \quad & E^\intercal \lambda = (1-\gamma)\mu + \gamma P^\intercal \lambda, \quad \lambda \geq 0.\end{aligned} \tag{Primal IL}$$

A parametric approach: Linear MDPs

Linear MDP [Jin et al., 2020]

We make the linear MDP assumption. That is, there exists 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

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

In the sequel, we will use the following compact matrix notation:

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

Remarks:

- The Linear MDP is a standard assumption in RL literature.
- The feature mapping Φ is assumed known.
- The variables w and M are unknown.

The constraint splitting trick

- We will now derive our algorithm, dubbed as $\texttt{P}^2\texttt{IL}$, using Primal IL.
- We use variable splitting to obtain advantageous (i.e., exact) and computable model-free policy updates.
- To begin, we plug in the (Linear MDP) structure in (Primal IL) as follows[1]

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

$$\Downarrow$$

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

- Supplementary slide 7 derives the inexact proximal point updates for λ and ρ on the Lagrangian.

[1] A similar trick appeared outside the imitation learning in [Mehta and Meyn, 2020], [Lee and He, 2019] and [Bas-Serrano et al., 2021]

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

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

Initialize π_0 as uniform distribution over $\mathcal{A}$
for $k = 1, \dots K$ **do**

`// Policy evaluation`

$$(w_k, \theta_k) \approx \underset{w \in \mathcal{W}, \theta \in \Theta}{\arg\max}\ \mathcal{G}_k(w, \theta)$$

`// Policy improvement`

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

end for

- $\mathcal{G}_k(w, \theta)$ is the following concave and smooth function:[2]

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

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

[2] This term is called the logistic Bellman error [Bas-Serrano et al., 2021].

Guarantees for $\texttt{P}^2\texttt{IL}$

- We consider errors in the maximization of $\mathcal{G}_k(w, \boldsymbol{\theta})$, i.e. $\epsilon_k = \mathcal{G}_k(w_k^\star, \theta_k^\star) - \mathcal{G}_k(w_k, \theta_k)$.
- We check how errors propagate and then control their size.

Error propagation

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

$$\max_{r \in \mathcal{R}} \langle \lambda_{\pi_\mathsf{E}} - \lambda_{\widehat{\pi}_K}, r \rangle \leq \frac{1}{K} \left(\log\left(m\,|\mathcal{A}|\right) + C \sum_k \sqrt{\epsilon_k} + \sum_k \epsilon_k \right).$$

Error control

Let (w_k, θ_k) be the output of the stochastic gradient ascent (SGA) 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 $1-\delta$.

Remarks:

- Choosing $K = \Omega(\epsilon^{-1})$ and $T = \Omega(\epsilon^{-4})$ we obtain $\mathcal{O}(\epsilon^{-5})$ online interactions.
- We use samples to approximate the gradients $\nabla_\theta \mathcal{G}_k$ and $\nabla_w \mathcal{G}_k$.
- We analyze the effect of the biased gradients in the SGA routine.

Guarantees for $\texttt{P}^2\texttt{IL}$

- We consider errors in the maximization of $\mathcal{G}_k(w, \boldsymbol{\theta})$, i.e. $\epsilon_k = \mathcal{G}_k(w_k^\star, \theta_k^\star) - \mathcal{G}_k(w_k, \theta_k)$.
- We check how errors propagate and then control their size.

Error propagation

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

$$\max_{r \in \mathcal{R}} \langle \lambda_{\pi_{\mathsf{E}}} - \lambda_{\widehat{\pi}_K}, r \rangle \leq \frac{1}{K}\left(\log\left(m\,|\mathcal{A}|\right) + C \sum_k \sqrt{\epsilon_k} + \sum_k \epsilon_k \right).$$

Error control

Let (w_k, θ_k) be the output of the stochastic gradient ascent (SGA) 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 $1-\delta$.

Remarks:

- Choosing $K = \Omega(\epsilon^{-1})$ and $T = \Omega(\epsilon^{-4})$ we obtain $\mathcal{O}(\epsilon^{-5})$ online interactions.
- We use samples to approximate the gradients $\nabla_\theta \mathcal{G}_k$ and $\nabla_w \mathcal{G}_k$.
- We analyze the effect of the biased gradients in the SGA routine.
- Please note the presence of the inconspicuous constant β in the denominator

Online IL experiments: Discrete actions

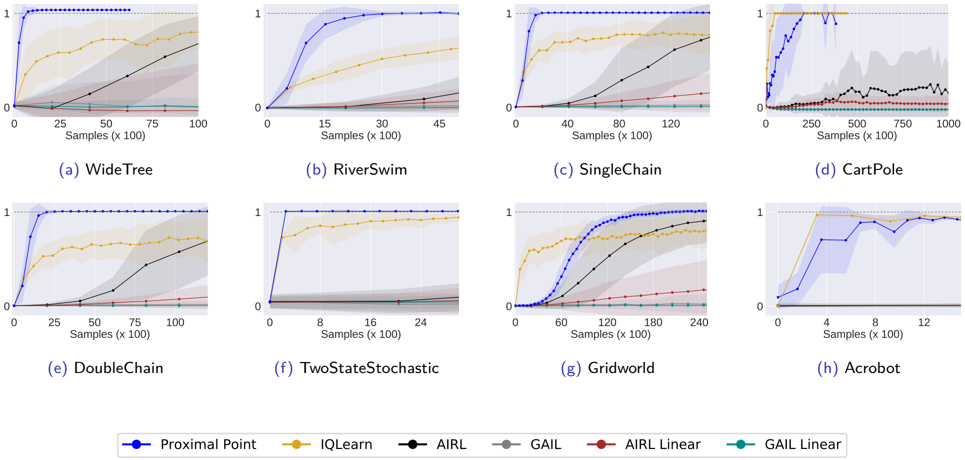
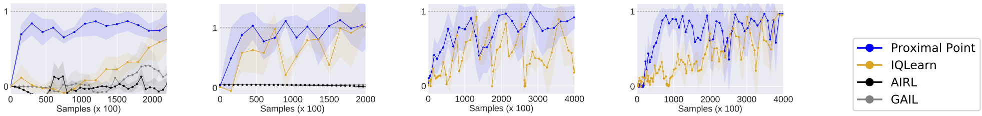


Figure: **Online IL Experiments**. We show the total returns vs the number of env steps.

Continuous control experiments



(a) HalfCheetah (b) Ant (c) Hopper (d) Walker2d

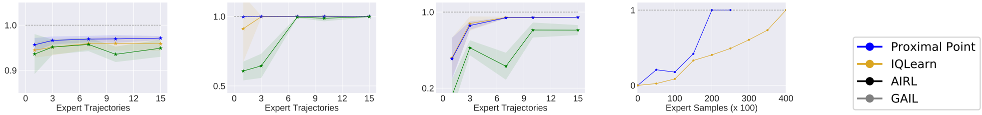
Figure: **Neural function approximation experiments.**

- This setting with non linear function approximation and continuous actions is not covered by theory.
- However the empirical performance is convincing vs IQLearn [Garg et al., 2021].

Offline experiments

- The algorithm works offline just changing the center point in the Bregman divergence

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

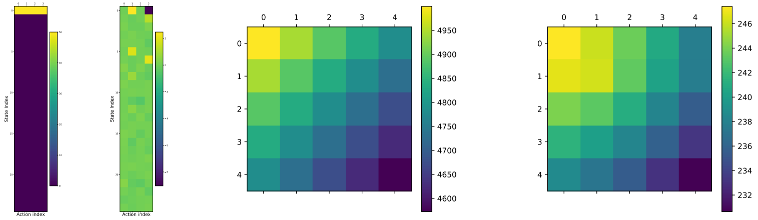


(a) Acrobot-v1 (b) CartPole-v1 (c) LunarLander-v2 (d) Pong

Figure: **Offline IL Experiments**

Recovered rewards

- In the policy evaluation step, we learn also a reward function.
- The recovered cost r^K is not similar to r_{true} (not surprising given reward shaping [Ng and Russell, 2000]).
- However, the value functions $V^\star_{\mathbf{c}_{\text{true}}}$ and $V^\star_{r^K}$ are $\implies$ We recover the optimal policy acting greedy wrt r^K.[3]



(a) r_{true} (b) r^K (c) $V^\star_{r_{\text{true}}}$ (d) $V^\star_{r^K}$

[3]We observed this empirically. Formal guarantees are an open question.

An exploration assumptions towards the sample complexity guarantees

◦ EULA: In infinite horizon Linear MDP, it is common to make the following assumption.

Positive definite covariance matrix

For any policy π^k generated during the iterations of the $\mathtt{P^2IL}$ algorithm, it holds that

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

Remarks: ◦ Roughly speaking, the assumption is rather strong.

◦ For example, if we use one hot features, the condition is equivalent to

$$\lambda_{\pi^k}(s,a) \geq \beta > 0 \quad \forall s,a \in \mathcal{S}\times\mathcal{A}, \quad \forall k \in [K].$$

◦ It means that all the policies π^k should visit all state action pairs with positive probability.

◦ In real scenarios, there are often states of the environment that should be avoided.

A new algorithm without exploration assumption [Viano et al., 2024]

◦ Let us recall our goal

The imitation learning goal

The goal is to learn an ϵ-optimal policy using as few resources as possible.

An ϵ-optimal policy with respect to the expert

A policy π is said ϵ-optimal policy with respect to the expert policy π_{E} if it satisfies the following:

$$(1-\gamma)\left\langle \mu, V^{\pi_{\mathsf{E}}}_{r_{\mathrm{true}}} - V^{\pi}_{r_{\mathrm{true}}} \right\rangle \leq \epsilon \quad \text{or} \quad \langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi}, r_{\mathrm{true}} \rangle \leq \epsilon.$$

Remarks:

◦ In the sequel, we will interpret $\sum_{k=1}^{K} \langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r_{\mathrm{true}} \rangle$ as *regret*

◦ We will use an online learning framework to optimize this regret

◦ We will need to overcome that a direct sublinear regret characterization wrt π_k is not possible:

- r_{true} is never observed by the learner
- The learner has no feedback on the decisions made.

Online learning: Basics

∘ In online linear minimization, a learner faces a non-stationary environment for K rounds.

Online linear minimization with full information

for $k = 1, \ldots, K$ **do**
The learner plays a decision x^k from a convex set $\mathcal{X}$.
The environment choose a loss vector ℓ^k.
The learner suffer a cost $\left\langle \ell^k, x^k \right\rangle$.
The learner observes the vector ℓ^k.
end for

Remarks: ∘ The object of interest in online learning is the *regret* against a comparator $x^\star \in \mathcal{X}$.

$$\mathrm{Regret}(K; x^\star) = \sum_{k=1}^{K} \left\langle \ell^k, x^k - x^\star \right\rangle$$

- ℓ^k is called the loss vector.
- x^k are the learner decisions.
- $x^\star$ is called the *comparator*.

∘ Typically, the comparator is chosen to maximize the regret (i.e., worst case).

A game theoretic approach

- We will need the following, trivial decomposition of the ϵ-approximate solution concept for IL:

$$\sum_{k=1}^{K} \langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r_{\text{true}} \rangle = \sum_{k=1}^{K} \left\langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r_{\text{true}} - r^k \right\rangle + \sum_{k=1}^{K} \left\langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r^k \right\rangle \leq \epsilon.$$

- Hence, we set up a game between two players.
  - The policy player updates the policy π^k.
  - The reward player updates the rewards $\{r^k\}_{k=1}^{K}$.
- We will generate sequences $\{\pi_k\}_{k=1}^{K}$ and $\{r^k\}_{k=1}^{K}$ such that both sums grow sublinearly.

Remarks:

- In $\mathtt{P}^2\mathtt{IL}$, we directly optimize for the worst case r
- In this new approach, we will optimize the key variables based on what we observe *online*

An online learning view on the game

- Regret for the reward player:

$$\sum_{k=1}^{K} \left\langle \lambda^{\pi_k} - \lambda^{\pi_{\mathsf{E}}}, r^k - r_{\mathrm{true}} \right\rangle$$

  - $\{r^k\}_{k=1}^K$ is the sequence of decision produced by the no-regret algorithm used to update the reward.
  - $\{\lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}\}_{k=1}^K$ is the sequence of (negated) loss vectors.
  - r_{true} is the comparator.

- Regret for the policy player:

$$\sum_{k=1}^{K} \left\langle -r^k, \lambda^{\pi_k} - \lambda^{\pi_{\mathsf{E}}} \right\rangle$$

  - $\{\lambda^{\pi_k}\}_{k=1}^K$ is the sequence of occupancy measures of the policies $\{\pi_k\}_{k=1}^K$.
  - The sequence $\{\pi_k\}_{k=1}^K$ is interpreted as the sequence of decisions of the algorithm.
  - $\{r^k\}_{k=1}^K$ is the sequence of (negated) loss vectors.
  - $\lambda^{\pi_{\mathsf{E}}}$ acts as comparator, i.e., the occupancy measure of the expert policy.
- Supplementary slide 13 derives our algorithm

The new algorithm: `ILARL`

- We call the resulting algorithm `ILARL`: Imitation Learning via Adversarial Reinforcement Learning.

Imitation Learning via Adversarial Reinforcement Learning: `ILARL`

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

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

`// Reward players update`

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

`// Policy players update`

Find an estimator-uncertainty pair (z^k, b^k) such that

$$\gamma \left| \phi(s,a)^T z^k - PV^{k-1}(s,a) \right| \leq b^k(s,a) \qquad \forall s, a \in \mathcal{S} \times \mathcal{A} \quad \text{with high probability.}$$

Update Q and V values

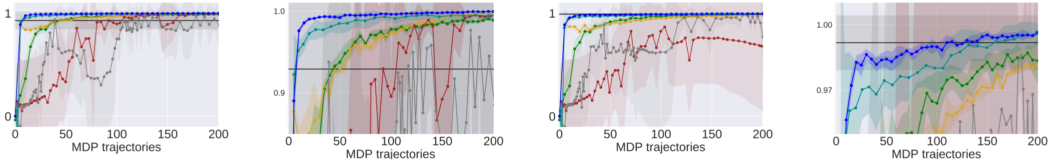
$$Q^k(s,a) = r^k(s,a) + \gamma\phi(s,a)^T z^k + b^k(s,a), \qquad V^k(s) = \left\langle \pi^k(a|s), Q^k(s,a) \right\rangle$$

Update policy

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

end for

Results with linear function approximation



(a) $N_E = 1$ (b) Detail for $N_E = 1$ (c) $N_E = 2$ (d) Detail for $N_E = 2$

Figure: Experiments on a continuous gridworld with a stochastic expert.The y-axis reports the normalized return. 1 correpsonds to the expert performance and 0 to the uniform policy one.

- This experiment shows that ILARL otperforms previous methods.
- Supplementary slide 18 explains possible extensions for deep learning.

A literature comparison

- $\mathtt{P}^2\mathtt{IL}$ is the first algorithm to jointly achieve convincing empirical performance, infinite horizon sample complexity guarantees and it avoids unstable alternated updates for cost and value function.
- ILARL is the first algorithm to achieve an online interaction bound without exploration assumption in infinite horizon linear MDP.

Algorithm	Sample Complexity Bound	Strong Empirical Performance	Training stability
Max Margin IRL	✗	✗	✓
Max Entropy IRL	✗	✗	✓
Max Likelihood IRL	✗	✓	✓
GAIL	✗	✓	✗
ASAF	✗	✓	✓
SQUIL	✗	✓	✓
ValueDICE	✗	✓	✗
Optimistic GAIL	✓	✗	✗
OAL	✓	✓	✗
IQLearn	✗	✓	✓
$\mathtt{P}^2\mathtt{IL}$	✓	✓	✓
ILARL	✓[4]	Stay tuned ...	Stay tuned ...

[4] without exploration assumptions

An additional comparison between theoretical imitation learning works.

Table: **Comparison with related algorithms** Our algorithms provide guarantees for the number of expert trajectories independent on $\mathcal{S}$ and Π without assumptions on the expert policy. For what concerns, the MDP trajectories we provide the best known results in finite and infinite horizon linear MDPs. By **Linear Expert**, me mean that the expert policy is $\pi(s) = \max_{a \in \mathcal{A}} \phi(s,a)^T \theta$ for some unknown vector θ.

Algorithm	Setting	Expert Traj.	MDP Traj.
Behavioural Cloning	Function Approximation, Offline [Agarwal et al., 2019]	$\mathcal{O}\left(\frac{H^4 \log\lvert\Pi\rvert}{\epsilon^2}\right)$	-
	Tabular, Offline [Rajaraman et al., 2020]	$\widetilde{\mathcal{O}}\left(\frac{H^2\lvert\mathcal{S}\rvert}{\epsilon}\right)$	-
	Linear Expert, Offline [Rajaraman et al., 2021]	$\widetilde{\mathcal{O}}\left(\frac{H^2 d}{\epsilon}\right)$	-
Mimic-MD [Rajaraman et al., 2020]	Tabular, Known Transitions, Deterministic Expert	$\mathcal{O}\left(\frac{H^{3/2}\lvert\mathcal{S}\rvert}{\epsilon}\right)$	-
OAL [Shani et al., 2021]	Tabular	$\mathcal{O}\left(\frac{H^2\lvert\mathcal{S}\rvert}{\epsilon^2}\right)$	$\mathcal{O}\left(\frac{H^4\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert}{\epsilon^2}\right)$
MB-TAIL [Xu et al., 2023]	Tabular, Deterministic Expert	$\mathcal{O}\left(\frac{H^{3/2}\lvert\mathcal{S}\rvert}{\epsilon}\right)$	$\mathcal{O}\left(\frac{H^3\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert}{\epsilon^2}\right)$
OGAIL [Liu et al., 2022b]	Linear Mixture MDP	$\mathcal{O}\left(\frac{H^3 d^2}{\epsilon^2}\right)$	$\mathcal{O}\left(\frac{H^4 d^3}{\epsilon^2}\right)$
PPIL [Viano et al., 2022]	Linear MDP, Persistent Excitation	$\mathcal{O}\left(\frac{d}{(1-\gamma)^2\epsilon^2}\right)$	$\mathcal{O}\left(\frac{d^2}{\beta^6(1-\gamma)^9\epsilon^5}\right)$
ILARL [Viano et al., 2024]	Linear MDP	$\mathcal{O}\left(\frac{d}{(1-\gamma)^2\epsilon^2}\right)$	$\mathcal{O}\left(\frac{d^3}{(1-\gamma)^8\epsilon^4}\right)$
ILARL (Finite Horizon) [Viano et al., 2024]	Episodic Linear MDP	$\mathcal{O}\left(\frac{dH^2}{\epsilon^2}\right)$	$\mathcal{O}\left(\frac{d^3 H^4}{\epsilon^2}\right)$

Remark : It can be shown that $\beta \geq d^{-1}$. So ILARL improves also the dimension dependence of $\mathtt{P}^2\mathtt{IL}$.

Side note: Towards an integrated analysis with neural function approximations

- Our analysis for P^{2}IL is limited to linear function approximation
- The promising results with DNNs call for a theoretical analysis beyond the linear setting
- Our recent work [Liu et al., 2022a] investigates Least Squares Value Iteration with DNNs
  - under ϵ-greedy exploration
  - achieves sublinear regret
- We consider general function spaces beyond the RKHS associated with the NTK regime
- As a result, we develop guidelines for architectures for practical deep RL
  - width or depth scaling
  - depending on the smoothness of the Q-function

Conclusions

More in the paper

- A detailed discussion on duality results for the linear programming formulation of imitation learning.
- Theoretical guarantees for the offline setting.
- Use the cost to generalize to new dynamics at test time.

Open questions

- Can we improve the sample complexity wrt to ϵ?
- Can we prove guarantees for the policy that acts greedly wrt the recovered cost?
- Can we analyze policy improvement errors as in [Geist et al., 2019]?
- Can we analyze also P^{2}IL with neural function approximation?

References I

[0] Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019).
Reinforcement learning: Theory and algorithms.
CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep.
(Cited on page 31.)

[0] Bas-Serrano, J., Curi, S., Krause, A., and Neu, G. (2021).
Logistic Q-learning.
In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
(Cited on pages 15 and 16.)

[0] Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. (2021).
IQ-learn: Inverse soft-Q learning for imitation.
In *Advances in Neural Information Processing Systems*.
(Cited on page 20.)

[0] Geist, M., Scherrer, B., and Pietquin, O. (2019).
A Theory of Regularized Markov Decision Processes.
In *International Conference on Machine Learning (ICML)*.
(Cited on page 33.)

[0] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018).
Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.
In *International Conference on Machine Learning*, pages 1856–1865.
(Cited on page 51.)

References II

[0] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020).
Provably efficient reinforcement learning with linear function approximation.
In *Conference on Learning Theory*, pages 2137–2143. PMLR.
(Cited on page 14.)

[0] Lee, D. and He, N. (2019).
Stochastic primal-dual q-learning algorithm for discounted mdps.
In *2019 american control conference (acc)*, pages 4897–4902. IEEE.
(Cited on page 15.)

[0] Liu, F., Viano, L., and Cevher, V. (2022a).
Understanding deep neural function approximation in reinforcement learning via $epsilon$-greedy exploration.
arXiv preprint arXiv:2209.07376.
(Cited on page 32.)

[0] Liu, Z., Zhang, Y., Fu, Z., Yang, Z., and Wang, Z. (2022b).
Learning from demonstration: Provably efficient adversarial policy imitation with linear function approximation.
In *International Conference on Machine Learning (ICML)*.
(Cited on page 31.)

[0] Mehta, P. G. and Meyn, S. P. (2020).
Convex Q-learning, Part 1: Deterministic optimal control.
arXiv:2008.03559.
(Cited on page 15.)

References III

[0] Ng, A. Y. and Russell, S. J. (2000).
Algorithms for inverse reinforcement learning.
In *International Conference on Machine Learning (ICML)*.
(Cited on page 22.)

[0] Rajaraman, N., Han, Y., Yang, L., Liu, J., Jiao, J., and Ramchandran, K. (2021).
On the value of interaction and function approximation in imitation learning.
Advances in Neural Information Processing Systems, 34:1325–1336.
(Cited on page 31.)

[0] Rajaraman, N., Yang, L., Jiao, J., and Ramchandran, K. (2020).
Toward the fundamental limits of imitation learning.
Advances in Neural Information Processing Systems, 33:2914–2924.
(Cited on page 31.)

[0] Rolland, P., Viano, L., Schuerhoff, N., Nikolov, B., and Cevher, V. (2022).
Identifiability and generalizability from multiple experts in inverse reinforcement learning.
Under review.
(Cited on pages 4 and 5.)

[0] Shani, L., Zahavy, T., and Mannor, S. (2021).
Online apprenticeship learning.
arXiv:2102.06924.
(Cited on page 31.)

References IV

[0] Sion, M. (1958).
On general minimax theorems.
Pacific Journal of Mathematics, 8(1):171–176.
(Cited on page 42.)

[0] Viano, L., Kamoutsi, A., Neu, G., Krawzuck, I., and Cevher, V. (2022).
Proximal point imitation learning.
Under review.
(Cited on pages 6 and 31.)

[0] Viano, L., Skoulakis, S., and Cevher, V. (2024).
Better imitation learning in discounted linear MDP.
(Cited on pages 24 and 31.)

[0] Xu, T., Li, Z., Yu, Y., and Luo, Z.-Q. (2023).
Provably efficient adversarial imitation learning with unknown transitions.
arXiv preprint arXiv:2306.06563.
(Cited on page 31.)

Supplementary Material

Strong Duality proof

$$(1-\gamma)\mathbb{E}_{s\sim\mu}[V^\pi(s)] = (1-\gamma)\,\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\mid s_0\sim\mu\right] \qquad \Rightarrow \text{ dual objective (D)}$$

$$= (1-\gamma)\sum_{s\in\mathcal{S},a\in\mathcal{A}}\sum_{t=0}^{\infty}\gamma^t\mathbb{P}(s_t=s,a_t=a\,|\,s_0\sim\mu,\,\pi)r(s,a)$$

$$= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\lambda^\pi(s,a)r(s,a) \qquad \Rightarrow \text{ primal objective (P)}$$

Proximal point method (PPM) in the Bregman setup

Definition: Bregman divergence

Let $\omega : \mathcal{X} \to \mathbb{R}$ be a distance generating function where ω is $1-$strongly convex w.r.t. some norm $\|\cdot\|$ on $\mathcal{X}$ and is continuously differentiable. The Bregman divergence induced by $\omega(\cdot)$ is given by

$$D_\omega(\mathbf{z}, \mathbf{z}') = \omega(\mathbf{z}) - \omega(\mathbf{z}') - \nabla\omega(\mathbf{z}')^\top(\mathbf{z} - \mathbf{z}').$$

∘ The proximal point method in the Bregman setup reads as follows:

$$\boxed{\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) + \frac{1}{\eta} D_\omega(\mathbf{x}, \mathbf{x}^k) \right\}}$$

Remarks:

- For example $\omega(\mathbf{x}) = \langle \mathbf{x}, \log \mathbf{x} \rangle$, gives the KL divergence.
- Avoids projection onto a simplex.
- Improves the dependence on the domain dimension.
- Forms the backbone of extra-gradient, mirror descent and others via inexact approximations.

Deriving $\mathtt{P}^2\mathtt{IL}$

- We derive the Lagrangian as follows

$$\min_{\rho\in\Delta^m,\lambda\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \underbrace{\max_{w\in\mathcal{W},\mathbf{V}\in\mathbb{R}^{|\mathcal{S}|},\theta\in\mathbb{R}^m} -\left\langle \rho - \Phi^T\widehat{\lambda^{\pi_E}}, w\right\rangle - \left\langle \mathbf{V}, -E^T\lambda + \gamma M^T\rho + (1-\gamma)\mu\right\rangle - \left\langle \theta, \Phi^T\lambda - \rho\right\rangle}_{\triangleq f(\rho,\lambda)}$$

- $\mathtt{P}^2\mathtt{IL}$ applies PPM to the above problem, i.e.

$$\rho_k,\lambda_k = \operatorname*{arg\,min}_{\rho\in\Delta^m,\lambda\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} f(\rho,\lambda) + \frac{1}{\eta}D(\rho,\Phi^T\lambda_{k-1}) + \frac{1}{\alpha}H(\lambda,\lambda_{k-1})$$

- $D(\cdot,\cdot)$ and $H(\cdot,\cdot)$ are respectively the relative entropy and the conditional relative entropy

Deriving $\texttt{P}^2\texttt{IL}$ (Continued)

- We can exchange $\max$ and $\min$ by Sion's theorem [Sion, 1958]

$$\max_{w\in\mathcal{W},\mathbf{V}\in\mathbb{R}^{|\mathcal{S}|},\theta\in\mathbb{R}^m}\min_{\rho\in\Delta^m,\lambda\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} -\left\langle \rho-\Phi^T\widehat{\lambda^{\pi_E}}, w\right\rangle - \left\langle \mathbf{V}, -E^T\lambda+\gamma M^T\rho+(1-\gamma)\mu\right\rangle - \left\langle \theta, \Phi^T\lambda-\rho\right\rangle$$
$$+\frac{1}{\eta}D(\rho,\Phi^T\lambda_{k-1})+\frac{1}{\alpha}H(\lambda,\lambda_{k-1})$$

- The minimizers of the inner minimization, i.e.

$$\rho_k^{w,\theta},\lambda_k^{\theta}=\arg\min_{\rho\in\Delta^m,\lambda\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} L(w,V,\theta,\rho,\lambda)$$

are known analytically.

Deriving $\texttt{P}^2\texttt{IL}$ (Continued)

- Indeed the maximizers are

$$\rho_k^{w,\theta}(i) \propto (\Phi^T \lambda_{k-1})(i)\, e^{\eta \delta^k_{w,\theta}(i)},$$
$$\pi_k^{\theta}(a|s) = \pi_{\lambda_{k-1}}(a|s)\, e^{\alpha (Q_\theta(s,a) - V_\theta^k(s))},$$
$$\lambda_k^{\theta} = \lambda^{\pi_k^\theta(a|s)}.$$

where we used the notation $\boldsymbol{\delta}^k_{w,\theta} \in \mathbb{R}^m$ by $\boldsymbol{\delta}^k_{w,\theta} := w + \gamma M V_\theta^k - \theta$, and we impose the following form for the value function to guarantee that π_k^θ lies in the simplex

$$V_\theta^k(s) = \frac{1}{\alpha} \log \left(\sum_a \pi_{\lambda_{k-1}}(a|s) e^{\alpha Q_\theta(s,a)} \right) \quad \text{where} \quad \mathbf{Q}_\theta = \Phi\theta$$

Deriving $\texttt{P}^2\texttt{IL}$ (Continued)

- It remains to find the maximizers of the outer maximization,

$$w_k^\star, \theta_k^\star = \underset{w \in \mathcal{W}, \theta \in \mathbb{R}^m}{\arg\max} \left\langle \rho_k^{w,\theta} - \Phi^T \widehat{\lambda^{\pi_E}}, w \right\rangle + \left\langle \mathbf{V}_\theta^k, -E^T \lambda_k^\theta + \gamma M^T \rho_k^{w,\theta} + (1-\gamma)\mu \right\rangle + \left\langle \theta, \Phi^T \lambda_k^\theta - \rho_k^{w,\theta} \right\rangle$$
$$+ \frac{1}{\eta} D(\rho_k^{w,\theta}, \Phi^T \lambda_{k-1}) + \frac{1}{\alpha} H(\lambda_k^\theta, \lambda_{k-1})$$
$$= \underset{w \in \mathcal{W}, \theta \in \mathbb{R}^m}{\arg\max} \mathcal{G}_k(w, \theta)$$

- $\mathcal{G}_k(w,\theta)$ is the following concave and smooth function.

$$\mathcal{G}_k(w,\theta) \triangleq -\frac{1}{\eta} \log \sum_{i=1}^{m} (\Phi^T \lambda_{k-1})(i) e^{-\eta \delta_{w,\theta}^k(i)} + (1-\gamma) \left\langle \mu, V_\theta^k \right\rangle - \left\langle \widehat{\lambda^{\pi_E}}, \Phi^T w \right\rangle.$$

- Then, the PPM update for the variables (ρ_k, λ_k) is given by

$$\lambda_k(i) \propto (\Phi^T \lambda_{k-1})(i)\, e^{-\eta \delta^k_{w_k^\star, \theta_k^\star}(i)},$$
$$\pi_k(a|s) \propto \pi_{\lambda_{k-1}}(a|s)\, e^{-\alpha Q_{\theta_k^\star}(s,a)},$$
$$\lambda_k = \lambda^{\pi_k(a|s)}$$

Deriving $\texttt{P}^2\texttt{IL}$ (Continued)

- It remains to find the maximizers of the outer maximization,

$$\begin{aligned} w_k^\star, \theta_k^\star &= \underset{w\in\mathcal{W},\theta\in\mathbb{R}^m}{\arg\max} \left\langle \rho_k^{w,\theta} - \Phi^T\widehat{\lambda^{\pi_E}}, w\right\rangle + \left\langle \mathbf{V}_\theta^k, -E^T\lambda_k^\theta + \gamma M^T\rho_k^{w,\theta} + (1-\gamma)\mu\right\rangle + \left\langle \theta, \Phi^T\lambda_k^\theta - \rho_k^{w,\theta}\right\rangle \\ &\qquad + \frac{1}{\eta}D(\rho_k^{w,\theta}, \Phi^T\lambda_{k-1}) + \frac{1}{\alpha}H(\lambda_k^\theta, \lambda_{k-1}) \\ &= \underset{w\in\mathcal{W},\theta\in\mathbb{R}^m}{\arg\max} \ \mathcal{G}_k(w,\theta) \end{aligned}$$

- $\mathcal{G}_k(w,\theta)$ is the following concave and smooth function.

$$\mathcal{G}_k(w,\theta) \triangleq -\frac{1}{\eta}\log\sum_{i=1}^m (\Phi^T\lambda_{k-1})(i)e^{-\eta\delta^k_{w,\theta}(i)} + (1-\gamma)\left\langle \mu, V_\theta^k\right\rangle - \left\langle \widehat{\lambda^{\pi_E}}, \Phi^T w\right\rangle.$$

- Then, the PPM update for the variables (ρ_k, λ_k) is given by

$$\begin{aligned} \lambda_k(i) &\propto (\Phi^T\lambda_{k-1})(i)\, e^{-\eta\delta^k_{w_k^\star,\theta_k^\star}(i)}, \\ \pi_k(a|s) &\propto \pi_{\lambda_{k-1}}(a|s)\, e^{-\alpha Q_{\theta_k^\star}(s,a)}, \\ \lambda_k &= \lambda^{\pi_k(a|s)} \end{aligned}$$

We are finally done (cf., Slide 14)!

Controlling the regret terms: the reward player

- If the class $\mathcal{R}$ is a convex set, then we can simply use Online Gradient Ascent for the reward player. That is,

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

- The caveat is that $\lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi^k}$ can not be computed because the dynamics are unknown.
- However, it is easy to obtain an unbiased estimate with bounded variance.

Controlling the regret terms: the policy player

- We develop a way to bound this term without exploration assumptions.
- $\forall \{Q^k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}_{k=1}^K$ and $\{V^k : \mathcal{S} \to \mathbb{R}$ s.t. $V^k(s) = \left\{\left\langle \pi^k(\cdot|s), Q^k(s,\cdot)\right\rangle\right\}_{k=1}^K\}$, we have

$$
\begin{aligned}
\sum_{k=1}^K \left\langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r^k \right\rangle = & \sum_{k=1}^K \mathbb{E}_{s\sim\lambda^{\pi_{\mathsf{E}}}} \left[\left\langle Q^k(s,\cdot), \pi_{\mathsf{E}}(s) - \pi^k(s)\right\rangle\right] && \text{(OMD)} \\
& + \sum_{k=1}^K \mathbb{E}_{s,a\sim\lambda^{\pi^k}} \left[Q^{k+1}(s,a) - r^k(s,a) - \gamma P V^k(s,a)\right] && \text{(Optimism 1)} \\
& + \sum_{k=1}^K \mathbb{E}_{s,a\sim\lambda^{\pi_{\mathsf{E}}}} \left[r^k(s,a) + \gamma P V^k(s,a) - Q^{k+1}(s,a)\right] && \text{(Optimism 2)} \\
& - \sum_{k=1}^K \mathbb{E}_{s,a\sim\lambda^{\pi^k}} \left[Q^{k+1}(s,a) - Q^k(s,a)\right] && \text{(Shift 1)} \\
& - \sum_{k=1}^K \mathbb{E}_{s,a\sim\lambda^{\pi_{\mathsf{E}}}} \left[Q^k(s,a) - Q^{k+1}(s,a)\right] && \text{(Shift 2)}
\end{aligned}
$$

Controlling each term

- (OMD) is sublinear in K if we update the policies via a no-regret algorithm.
- For example, we can use online mirror ascent with entropy as regularizer, i.e.

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

- (Shift 2) simply telescopes.
- (Shift 1) is small because the sequence of policies $\{\pi_k\}_{k=1}^K$ is slowly changing, i.e.

$$\max_{s \in \mathcal{S}} \left\| \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s) \right\|_1 \leq \mathcal{O}(\eta)$$

- With this observation, we have that

$$\sum_{k=1}^K \left\langle \lambda^{\pi_{\mathsf{E}}} - \lambda^{\pi_k}, r^k \right\rangle = o(K) + \sum_{k=1}^K \mathbb{E}_{s,a \sim \lambda^{\pi^k}} \left[Q^{k+1}(s,a) - r^k(s,a) - \gamma P V^k(s,a) \right] \qquad \text{(Optimism 1)}$$

$$+ \sum_{k=1}^K \mathbb{E}_{s,a \sim \lambda^{\pi_{\mathsf{E}}}} \left[r^k(s,a) + \gamma P V^k(s,a) - Q^{k+1}(s,a) \right] \qquad \text{(Optimism 2)}$$

Controlling each term (Continued)

- We are left with controlling (Optimism 1) and (Optimism 2).
- If the transition were known, we could make the terms zero by the following update rule

$$\begin{aligned} Q^{k+1}(s,a) &= r^k(s,a) + \gamma P V^k(s,a) \\ &= r^k(s,a) + \gamma P^{\pi^k} Q^k(s,a). \end{aligned}$$

- That is applying the Bellman evaluation operator of the policy π^k on Q^k.
- Unfortunately, this can not be done because we do not know the transition dynamics, i.e. the matrix P.
- We circumvent the problem finding an estimator-uncertainty pair (θ^k, b^k) such that

$$\gamma \left| \phi(s,a)^T \theta^k - P V^k(s,a) \right| \leq b^k(s,a) \qquad \forall s, a \in \mathcal{S} \times \mathcal{A}$$

with high probability.

Controlling each term (Continued)

◦ We use the estimator-uncertainty uncertainty pair to approximate the update

$$r^k(s,a) + \gamma P V^k(s,a)$$

as

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

It follows that with high probability,

- (Optimism 2) ≤ 0
- (Optimism 1) $\leq 2 \sum_{k=1}^K \mathbb{E}_{s,a \sim \lambda^{\pi^k}} \left[b^k(s,a) \right]$

◦ In the paper, we show how to design uncertainties $\{b^k\}_{k=1}^K$ such that

$$2 \sum_{k=1}^K \mathbb{E}_{s,a \sim \lambda^{\pi^k}} \left[b^k(s,a) \right] = o(K)$$

without requiring exploration assumptions at all!

◦ Our algorithm can be found in Slide 25

Take Aways for Deep Imitation Learning.

◦ The improved result follows using policies in the form

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

where $Q^k(s,a)$ is an upper bound on $r^k(s,a) + \gamma PV^k(s,a)$.

- Going beyond linear functions, we can instantiate a neural network $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ trying to predict $y^k(s,a) = r^k(s,a) + \gamma PV^k(s,a)$.
- Moreover, we can try heuristics to estimate the confidence interval width $\Delta(s,a)$ of the neural network prediction $f(s,a)$.
- Therefore, we can use updates

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

- If the environmnet has continuous actions, these updates can be approximated via Soft Actor Critic [Haarnoja et al., 2018].