Imitation Learning without exploration assumptions

Luca Viano, Stratis Skoulakis and Volkan Cevher

EPFL

A new algorithm without exploration assumption

∘ Let us recall our goal

ϵ-optimal policy w.r.t. the expert

A policy π is said ϵ-optimal policy if

$$\frac{1}{1-\gamma}\langle\lambda^{\pi_{\mathrm{E}}}, r_{\mathrm{true}}\rangle - \frac{1}{1-\gamma}\langle\lambda^{\pi}, r_{\mathrm{true}}\rangle \leq \epsilon$$

∘ We can play this trick for a sequence $\{r^k\}_{k=1}^K$ to be chosen later.

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

∘ If both sums grows sublinearly, the policy $\pi_{\mathrm{out}} \sim \mathrm{Unif}(\{\pi_k\}_{k=1}^K)$ is ϵ-optimal for K large enough.

∘ Therefore, we aim at generating sequences $\{\pi_k\}_{k=1}^K$ and $\{r^k\}_{k=1}^K$ such that both sums grow sublinearly.

An online learning view

∘ We can interpret the two sums as two sources of regret.

Regret for the reward player

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

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

Regret for the policy player

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

- $\{\lambda^{\pi_k}\}_{k=1}^K$ is the sequence of occupancy meaures 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.
- $\lambda^{\pi_{\mathrm{E}}}$ acts as comparator, i.e. the occupancy measure of the expert policy.

Controlling the regret terms: the policy player

∘ We develop a way to bound this term without exploration assumptions.

∘ For any sequence $\{Q^k : \mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}\}_{k=1}^K$ and $\{V^k : \mathcal{S}\rightarrow\mathbb{R}$ such that $V^k(s) = \left\{\left\langle\pi(\cdot|s), Q^k(s,\cdot)\right\rangle\right\}_{k=1}^K$ is the following

$$\sum_{k=1}^K\left\langle\lambda^{\pi_{\mathrm{E}}} - \lambda^{\pi_k}, r^k\right\rangle = \sum_{k=1}^K\mathbb{E}_{s\sim d^{\pi_{\mathrm{E}}}}\left[\left\langle Q^k(s,\cdot), \pi_{\mathrm{E}}(s) - \pi^k(s)\right\rangle\right] \qquad \text{(OMD)}$$

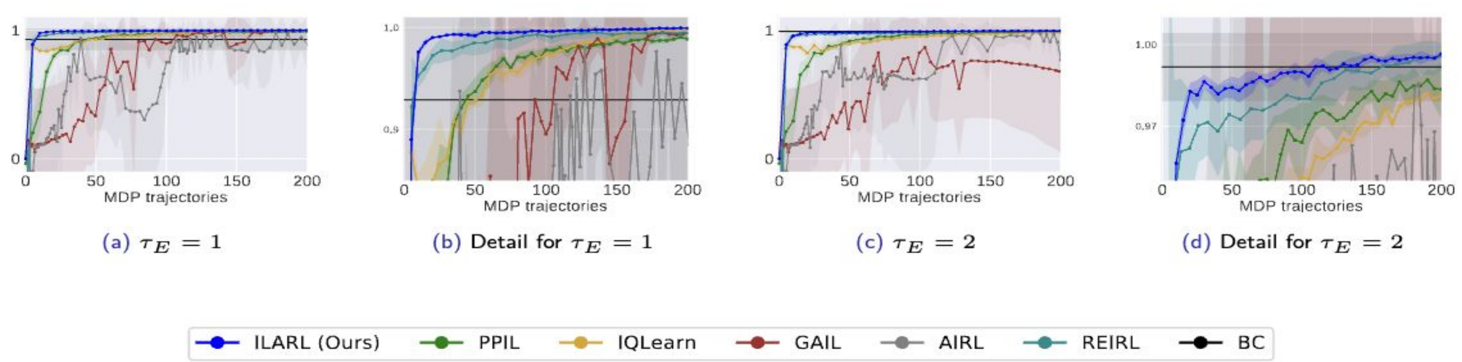
$$+\sum_{k=1}^K\mathbb{E}_{s,a\sim d^{\pi^k}}\left[Q^{k+1}(s,a) - r^k(s,a) - \gamma PV^k(s,a)\right] \qquad \text{(Optimism 1)}$$

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

$$-\sum_{k=1}^K\mathbb{E}_{s,a\sim d^{\pi^k}}\left[Q^{k+1}(s,a) - Q^k(s,a)\right] \qquad \text{(Shift 1)}$$

$$-\sum_{k=1}^K\mathbb{E}_{s,a\sim d^{\pi_{\mathrm{E}}}}\left[Q^k(s,a) - Q^{k+1}(s,a)\right] \qquad \text{(Shift 2)}$$

Results with linear function approximation



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

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.

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.

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

∘ With this observation, we have that

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

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

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,

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

∘ The caveat is that $\lambda^{\pi_{\mathrm{E}}} - \lambda^{\pi^k}$ can not be computed because the dynamics are unknown.

∘ However, it is easy to obtain an unbiased bounded variance estimate.

Controlling each term (Continued)

∘ We are left with controlling (Optimism 1) and (Optimism 2).

∘ If the transition where known, we could make the terms zero by the following update rule

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

∘ 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

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

with high probability.

Controlling each term (Continued)

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

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

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

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

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

without requiring exploration assumptions at all !

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 $f : \mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}$ 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].

The new algorithm: ILARL

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

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

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

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**