Bilevel Reinforcement Learning via the Development of Hyper-gradient without Lower-Level Convexity

Yan Yang

CSML 2025, Beijing

Academy of Mathematics and Systems Science
Chinese Academy of Sciences
University of Chinese Academy of Sciences

Joint work with **Dr. Bin Gao** and **Prof. Ya-xiang Yuan**

Table of contents

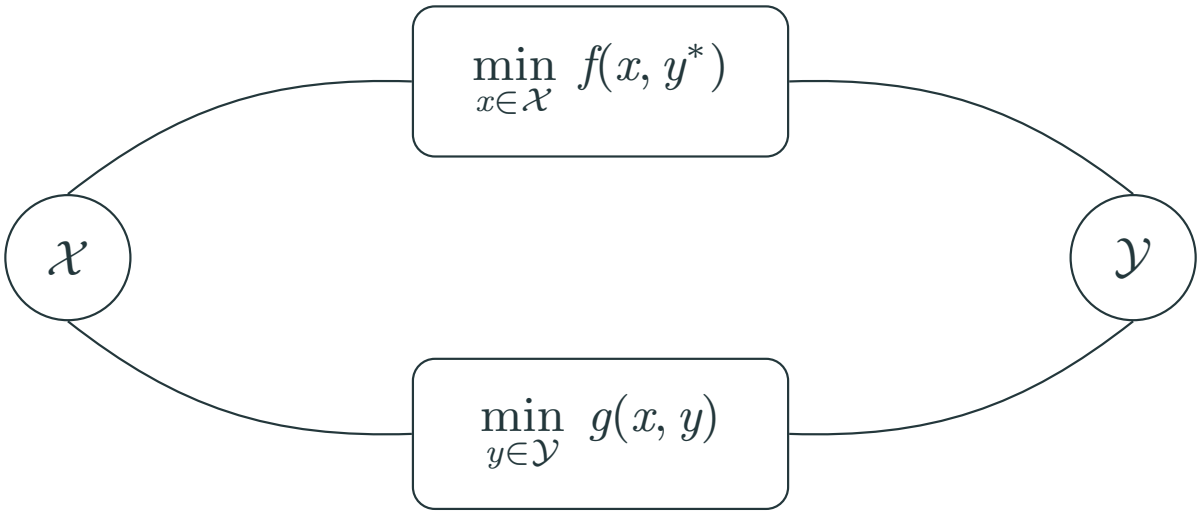
Bilevel Optimization and Hyper-gradient

Standard formulation

Bilevel Optimization (BiO)

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x, y^\star) \\ \text{s.t.} \quad & y^\star \in \arg\min_{y \in \mathcal{Y}} \; g(x, y) \end{aligned}$$

The nested structure couples the **upper level** and **lower level**



Applications

Model selection [Kunapuli et al., 2008; Giovannelli et al., 2021]

Hyper-parameters optimization [Franceschi et al., 2018; Bao et al., 2021]

Data poisoning [Liu et al., 2024]

Reinforcement learning [Hong et al., 2023; Chakraborty et al., 2024; Shen et al., 2024]

. . .

Three visits to hyper-gradient

Recall the BiO problem

$$\min_{x \in \mathbb{R}^{d_x}} \quad \phi(x) := f(x, y^\star(x))$$

$$\text{s.t.} \quad y^\star(x) = \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y)$$

How to study the hyper-gradient $\nabla \phi$?

Approximate implicit differentia tion (AID)

Scenario of interest: $g(x, \cdot)$ is strongly convex

$$y^*(x) = \arg\min_{y \in \mathbb{R}^{d_y}} g(x, y)$$

By optimality condition & implicit function theorem

$$0 \equiv \nabla_y g(x, y^*(x)) \Rightarrow \nabla^2_{xy} g(x, y^*(x)) + \nabla_x y^*(x)^\top \nabla^2_{yy} g(x, y^*(x)) = 0$$

Hyper-gradient computation

$$\begin{aligned}
\nabla \phi(x) &:= \nabla f(x, y^\star(x)) \\
&= \nabla_x f(x, y^\star(x)) + \nabla_x y^\star(x)^\top \nabla_y f(x, y^\star(x)) \\
&= \nabla_x f(x, y^\star) - \nabla^2_{xy} g(x, y^\star) \left[\nabla^2_{yy} g(x, y^\star)\right]^{-1} \nabla_y f(x, y^\star)
\end{aligned}$$

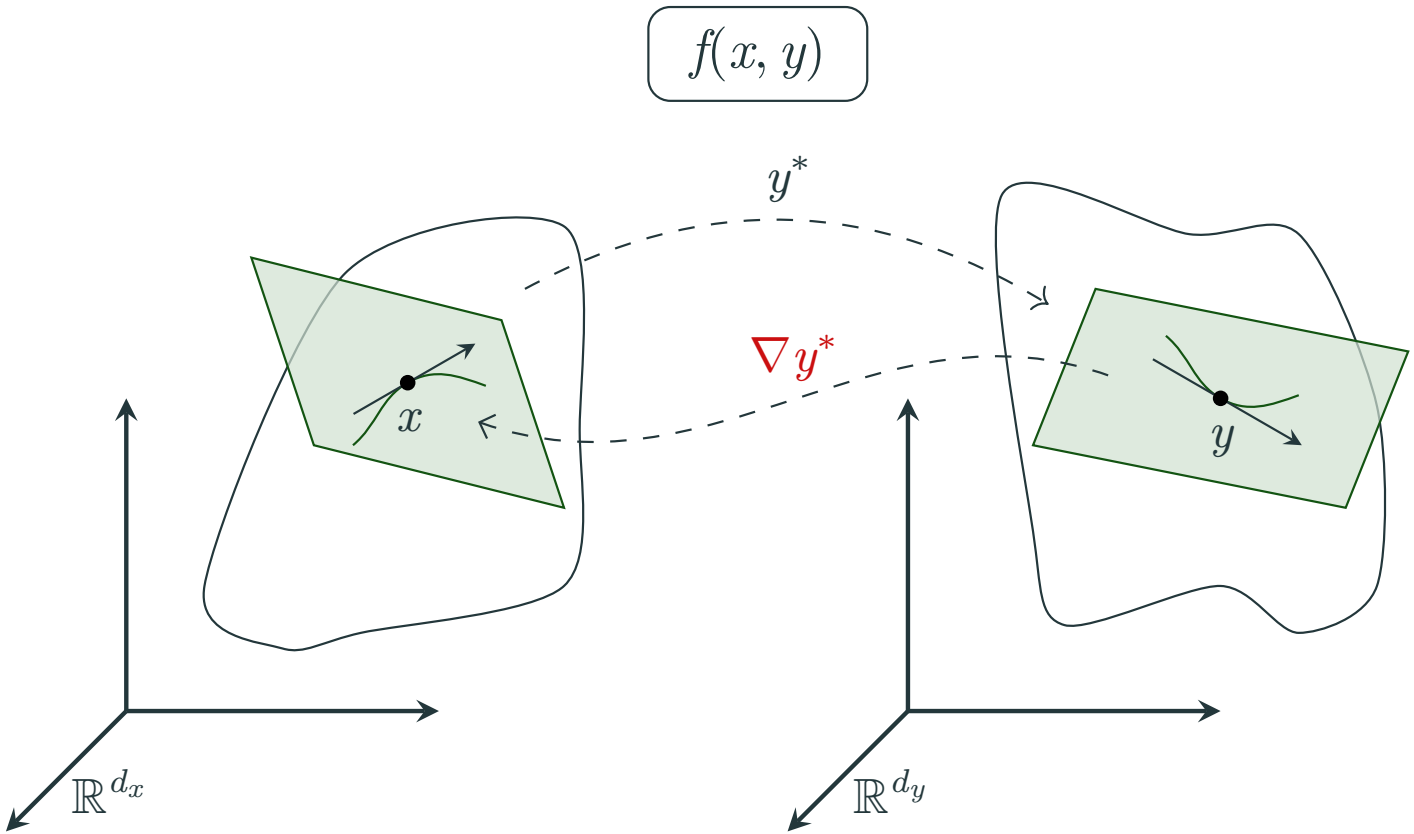
Vanilla update rule

$$\mathbf{1}\times: x^+ = x - \beta \left(\nabla_x f(x, y) - \nabla^2_{xy} g(x, y) \left[\nabla^2_{yy} g(x, y)\right]^{-1} \nabla_y f(x, y) \right)$$

$$\mathbf{N}\times: y^+ = y - \alpha \nabla_y g(x, y)$$

First visit to hyper-gradient

Geometry



Algebra

$$\nabla \phi(x) = \nabla_x f(x, y^\star) - \nabla^2_{xy} g\left(x, y^\star\right) \left[\nabla^2_{yy} g\left(x, y^\star\right)\right]^{-1} \nabla_y f(x, y^\star)$$

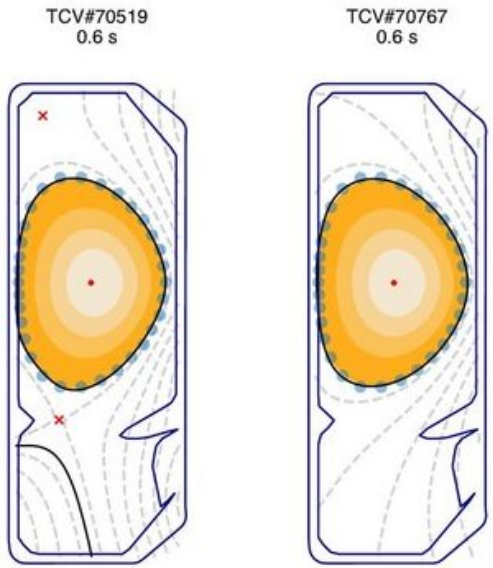
Bilevel Reinforcement Learning

Reinforcement learning (RL)

Agents learn to make sequential decisions, by **interacting** with environments

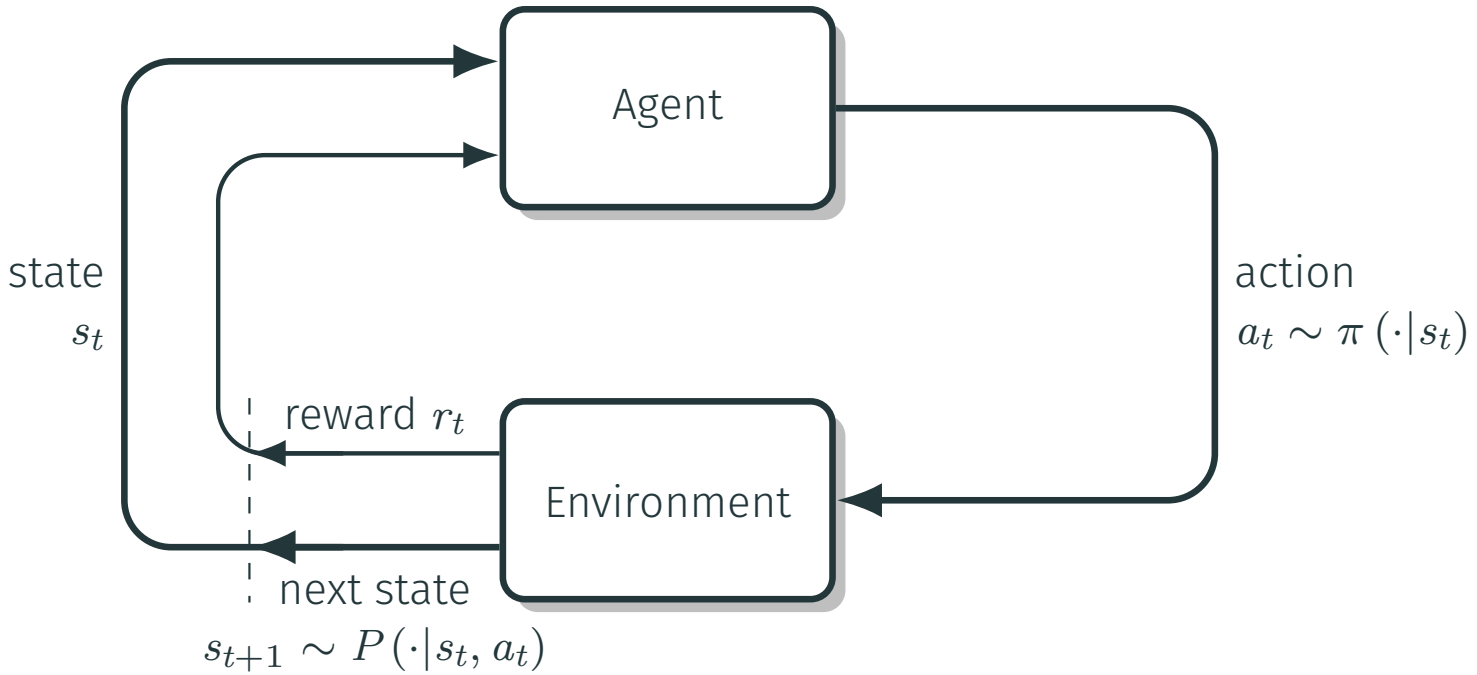
Silver et al., 2016
Nature

Samvelyan et al., 2019
DeepMind



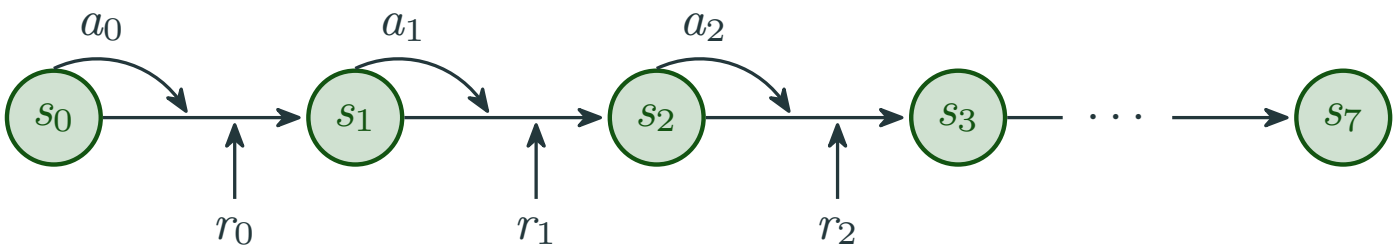
Degrave et al., 2022
Nature

Markov decision process (MDP)



- Environment $\mathcal{M}_\tau = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \tau)$
- State space $\mathcal{S}$, action space $\mathcal{A}$
- Transition matrix $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$
- Reward function $r \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$
- Policy $\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, which induces $P^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$: $P^\pi_{ss'} = \sum_a \pi_{sa} P_{sas'}$

Soft value function and Q-value function



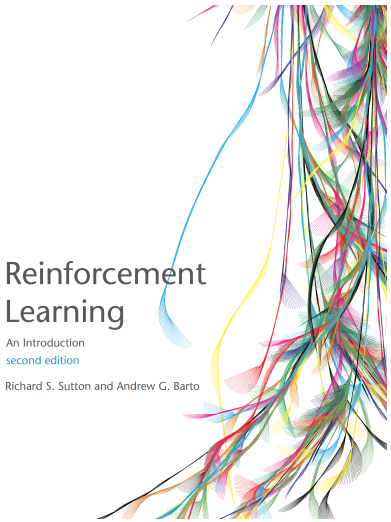
$$\forall s \in \mathcal{S}: \qquad V_s^\pi := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_{s_t a_t} + \tau h(\pi_{s_t})\right) \;\middle|\; s_0 = s, \pi, \mathcal{M}_\tau\right]$$

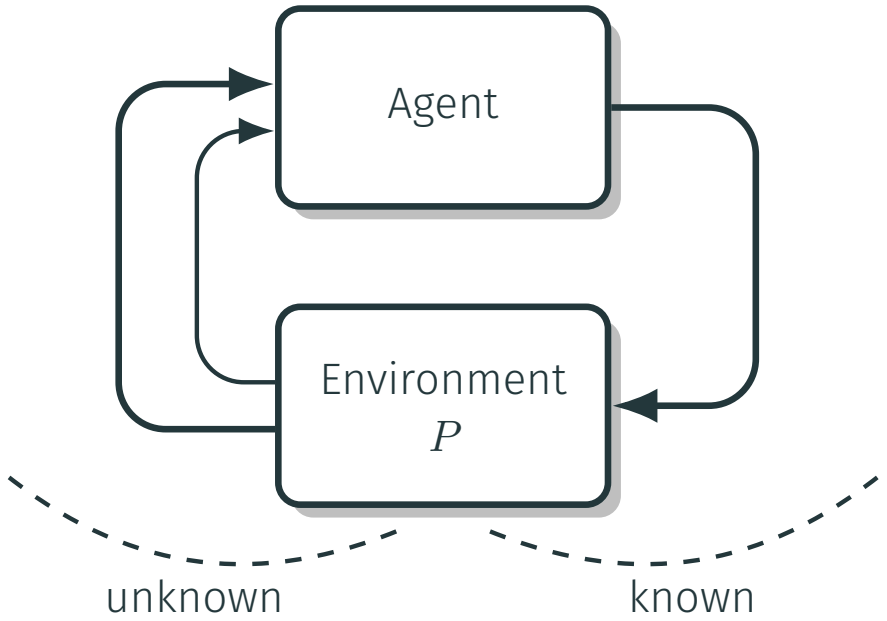
$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \qquad Q_{sa}^\pi := r_{sa} + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[V_{s'}^\pi\right]$$

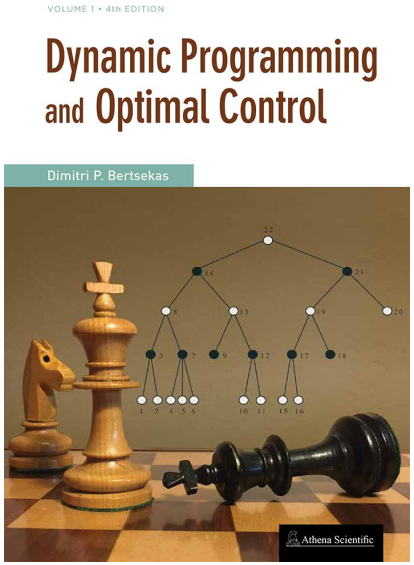
Given the initial state distribution ρ, RL aims to solve

$$\begin{aligned}\max_\pi \quad V^\pi(\rho) &:= \mathbb{E}_{s\sim\rho}\left[V_s^\pi\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_{s_t a_t} + \tau h(\pi_{s_t})\right) \;\middle|\; s_0 \sim \rho, \pi, \mathcal{M}_\tau\right]\end{aligned}$$

Two scenarios







Model-free

Model-based

Bilevel RL formulation

Parameterized MDP: $\mathcal{M}_\tau(x) = (\mathcal{S}, \mathcal{A}, P, r(x), \gamma, \tau)$

Quantities in bilevel context

$$\forall s \in \mathcal{S}: \qquad V_s^\pi(x) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_{s_t a_t}(x) + \tau h(\pi_{s_t})\right) \;\middle|\; s_0 = s, \pi, \mathcal{M}_\tau(x)\right]$$

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \qquad Q_{sa}^\pi(x) := r_{sa}(x) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[V_{s'}^\pi(x)\right]$$

Bilevel RL problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg\min_\pi - V_{\mathcal{M}_\tau(x)}^\pi(\rho) \end{aligned}$$

Applications of bilevel RL

Reward shaping [Hu et al., 2020; Devidze et al., 2022]

$$\min_{x} -V_{\bar{\mathcal{M}}_{\bar{\tau}}}^{\pi^*(x)}(\bar{\rho}),\ \text{s.t. } \pi^*(x) = \arg\min_{\pi \in \Delta^{|\mathcal{A}|}} -V_{\mathcal{M}_\tau(x)}^{\pi}(\rho)$$

RL from human feedback (RLHF) [Christiano et al., 2017]

$$\begin{aligned} \min_{x} \quad & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))}\left[l_h(d_1, d_2, y; x)\right] \\ \text{s.t.} \quad & \pi^*(x) = \arg\min_{\pi} -V_{\mathcal{M}_\tau(x)}^{\pi}(\rho) \end{aligned}$$

- Trajectory sampled by $\pi^*(x)$: $d_i = \{(s_h^i, a_h^i)\}_{h=0}^{H-1}\ (i = 1, 2)$
- Preference for d_1 over d_2: $y \in \{0, 1\}$

More applications

- Apprenticeship learning [Arora and Doshi, 2021]
- Contract design [Wu et al., 2024]
- Robot navigation [Chakraborty et al., 2024]

Problem formulation

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s. t.} \quad & \pi^*(x) = \arg\min_{\pi} -V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) \end{aligned}$$

where $V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(r_{s_t a_t}(x) + \tau h(\pi_{s_t})\right) \,\middle|\, s_0 \sim \rho, \pi, \mathcal{M}_\tau(x)\right]$

Main difficulties

- Lower-level inherent **non-convexity** [Agarwal et al., 2020; Lan, 2023 MP]
- Only the **non-uniform PL** condition [Mei et al., 2020]
- **Existence** of hyper-gradient $\nabla\phi(x)$? [Shen et al., 2024]

Related work

- An AID-based bilevel RL framework, PARL [Chakraborty et al., ICLR 2024]
- A penalty-based method, PBRL [Shen et al., ICML 2024]
- A stochastic bilevel RL method, HPGD [Thoma et al., NeurIPS 2024]

Comparison among provable bilevel RL algorithms

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(\log \epsilon^{-1}\right)$	1st+2nd
PBRL	Deter.	$\mathcal{O}(\lambda\epsilon^{-1})$	$\mathcal{O}\left(\log \lambda^2\epsilon^{-1}\right)$	1st
HPGD	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}\left(\log \epsilon^{-1}\right)$	1st
M-SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(1\right)$	1st
SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(\log \epsilon^{-1}\right)$	1st
Stoc-SoBiRL	Stoch.	$\widetilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}\left(\log \epsilon^{-1}\right)$	1st

Model-based Soft BiRL

The fixed-point equation

Revisit the strong convexity requirement in AID

- Unique lower-level solution $\Longrightarrow y^*(x)$
- Nonsingular Hessian $\Longrightarrow \nabla y^*(x)$

Softmax temporal value consistency [Nachum et al., 2017]

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad \pi^*_{sa}(x) = \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V^*_{s'}(x) - V^*_s(x)\right)\right)$$

$$\forall s \in \mathcal{S}: \quad V^*_s(x) = \tau \log\left(\sum_a \exp\left(\frac{r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V^*_{s'}(x)}{\tau}\right)\right)$$

Take the derivative

$$\nabla \pi^*(x) = \tau^{-1} \mathrm{diag}\left(\pi^*(x)\right)\left(\nabla r(x) - U \nabla V^*(x)\right)$$

The fixed-point equation

Revisit the strong convexity requirement in AID

- Unique lower-level solution $\Longrightarrow y^*(x)$
- Nonsingular Hessian $\Longrightarrow \nabla y^*(x)$

Softmax temporal value consistency [Nachum et al., 2017]

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad \pi^*_{sa}(x) = \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V^*_{s'}(x) - V^*_s(x)\right)\right)$$

$$\forall s \in \mathcal{S}: \quad V^*_s(x) = \tau \log\left(\sum_a \exp\left(\frac{r_{sa}(x) + \gamma \sum_{s'} P_{sas'} V^*_{s'}(x)}{\tau}\right)\right)$$

Take the derivative

$$\nabla \pi^*(x) = \tau^{-1} \mathrm{diag}\left(\pi^*(x)\right)\left(\nabla r(x) - U \nabla V^*(x)\right)$$

How to tackle $\nabla V^*(x)$?

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log\left(\exp\left(\frac{r(x) + \gamma P v}{\tau}\right) \cdot \mathbf{1}\right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}{\sum_a \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}$$

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log\left(\exp\left(\frac{r(x) + \gamma P v}{\tau}\right) \cdot \mathbf{1}\right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}{\sum_a \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}$$

$$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \Longrightarrow \text{unique solution } V^*(x)$$

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log\left(\exp\left(\frac{r(x) + \gamma P v}{\tau}\right) \cdot \mathbf{1}\right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}{\sum_a \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}$$

$$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \Longrightarrow \text{unique solution } V^*(x)$$

- Differentiate both sides of $\varphi(x, V^*(x)) = V^*(x)$

$$\nabla V^*(x) = \left(I - \nabla_v \varphi(x, V^*(x))\right)^{-1} \nabla_x \varphi(x, V^*(x))$$

Investigating $\nabla V^*(x)$

Consider the parameterized mapping $\varphi : \mathbb{R}^n \times \mathbb{R}^{|\mathcal{S}|} \mapsto \mathbb{R}^{|\mathcal{S}|}$

$$\varphi(x, v) = \tau \log\left(\exp\left(\frac{r(x) + \gamma P v}{\tau}\right) \cdot \mathbf{1}\right)$$

- Take the derivative

$$\frac{\partial \varphi_s(x, v)}{\partial v_{s'}} = \gamma \frac{\sum_a P_{sas'} \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}{\sum_a \exp\left(\tau^{-1}\left(r_{sa}(x) + \gamma \sum_{s''} P_{sas''} v_{s''}\right)\right)}$$

$\|\nabla_v \varphi(x, v)\|_\infty = \gamma < 1 \Longrightarrow$ unique solution $V^*(x)$

- Differentiate both sides of $\varphi(x, V^*(x)) = V^*(x)$

$$\nabla V^*(x) = \left(I - \nabla_v \varphi(x, V^*(x))\right)^{-1} \nabla_x \varphi(x, V^*(x))$$

An observation in MDP $\nabla_v \varphi\left(x, V^*(x)\right) = \gamma P^{\pi^*(x)}$

Designing the model-based algorithm

Model-based hyper-gradient

$$
\begin{aligned}
\nabla\phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla\pi^*(x)^\top \nabla_\pi f(x, \pi^*(x)) \\
&= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \left(\nabla r(x) - U \nabla V^*(x)\right)^\top \operatorname{diag}\left(\pi^*(x)\right) \nabla_\pi f(x, \pi^*(x)) \\
&= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \operatorname{diag}\left(\pi^*(x)\right) \nabla_\pi f(x, \pi^*(x)) \\
&\quad - \tau^{-1} \nabla_x \varphi\left(x, V^*(x)\right)^\top \left(I - \gamma P^{\pi^*(x)}\right)^{-\top} U^\top \operatorname{diag}\left(\pi^*(x)\right) \nabla_\pi f(x, \pi^*(x))
\end{aligned}
$$

Designing the model-based algorithm

Model-based hyper-gradient

$$
\begin{aligned}
\nabla\phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla\pi^*(x)^\top \nabla_\pi f(x, \pi^*(x)) \\
&= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \left(\nabla r(x) - U\nabla V^*(x)\right)^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\
&= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\
&\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)}\right)^{-\top} U^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x))
\end{aligned}
$$

Computing $\nabla\phi(x)$ is expensive!

Designing the model-based algorithm

Model-based hyper-gradient

$$\begin{aligned}\nabla\phi(x) &= \nabla_x f(x, \pi^*(x)) + \nabla\pi^*(x)^\top \nabla_\pi f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \left(\nabla r(x) - U\nabla V^*(x)\right)^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\ &= \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\ &\quad - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)}\right)^{-\top} U^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x))\end{aligned}$$

Computing $\nabla\phi(x)$ is expensive!

Inexact strategy

- $V_k^* := V^*(x_k) \longleftarrow V_k \quad Q_k^* := Q^*(x_k) \longleftarrow Q_k \quad \pi_k^* := \pi^*(x_k) \longleftarrow \pi_k$
- $A_k^{-1} b_k \longleftarrow w_k$ with $A_k := (I - \gamma P^{\pi_k})^\top \quad b_k := U^\top \operatorname{diag}(\pi_k) \nabla_\pi f(x_k, \pi_k)$

Amortization and approximation

Model-based hyper-gradient estimator

$$\widehat{\nabla}\phi\left(x_k, \pi_k, V_k, w_k\right) := \nabla_x f(x_k, \pi_k) + \frac{1}{\tau}\nabla r(x_k)^\top \operatorname{diag}\left(\pi_k\right)\nabla_\pi f(x_k, \pi_k) - \frac{1}{\tau}\nabla_x \varphi(x_k, V_k)^\top w_k$$

Soft Bellman operator solves lower-level problem

$$\mathcal{T}_{\mathcal{M}_\tau(x)}(Q)(s,a) = r_{sa}(x) + \gamma \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)}\left[\tau \log\left(\left\|\exp\left(Q\left(s', \cdot\right)/\tau\right)\right\|_1\right)\right]$$

Softmax temporal value consistency points to V and π

$$V_s^*\left(x\right) = \tau \log\left(\sum\nolimits_a \exp\left(Q_{sa}^*\left(x\right)/\tau\right)\right)$$

$$\pi_{sa}^*(x) = \frac{\exp\left\{Q_{sa}^*\left(x\right)/\tau\right\}}{\sum_{a'}\exp\left(Q_{sa'}^*\left(x\right)/\tau\right)}$$

Least squared perspective

$$w_k \approx \min_{w\in\mathbb{R}^{|\mathcal{S}|}} \frac{1}{2}\left\|A_k w - b_k\right\|^2$$

Model-based bilevel RL algorithm: M-SoBiRL

M-SoBiRL

$$\mathbf{1}\times: w_k = w_{k-1} - \xi\left(\left(A_k^\top A_k\right) w_{k-1} - A_k^\top b_k\right)$$

$$\mathbf{1}\times: V_k(s) = \tau \log\left(\sum\nolimits_a \exp\left(Q_k(s,a)/\tau\right)\right)$$

$$\mathbf{1}\times: x_{k+1} = x_k - \beta\widehat{\nabla}\phi\left(x_k, \pi_k, V_k, w_k\right)$$

$$\mathbf{N}\times: Q_{k,n} = \mathcal{T}_{\mathcal{M}_\tau(x_{k+1})}(Q_{k,n-1})$$

$$\mathbf{1}\times: \pi_{k+1} = \texttt{softmax}\left(Q_{k+1}/\tau\right)$$

Model-free Soft BiRL

Problem formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \phi(x) := f(x, \pi^*(x)) \\ \text{s.t.} \quad & \pi^*(x) = \arg\min_{\pi} -V^{\pi}_{\mathcal{M}_\tau(x)}(\rho) \end{aligned}$$

with $f(x, \pi) = \mathbb{E}_{d_i \sim \rho(d;\pi)} \left[l(d_1, d_2, ..., d_I; x) \right]$

Main difficulties

$$\begin{aligned} \nabla\phi(x) = \nabla_x f(x, \pi^*(x)) + \tau^{-1} \nabla r(x)^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \\ - \tau^{-1} \nabla_x \varphi(x, V^*(x))^\top \left(I - \gamma P^{\pi^*(x)} \right)^{-\top} U^\top \operatorname{diag}(\pi^*(x)) \nabla_\pi f(x, \pi^*(x)) \end{aligned}$$

☹ Relying explicitly on the black-box P

☹ Involving large-scale matrix multiplications

Absorb P into an expectation

$$
\begin{aligned}
\nabla V_s^* (x) &= \left(I - \nabla_v \varphi \left(x, V^* (x)\right)\right)_s^{-1} \nabla_x \varphi \left(x, V^* (x)\right) \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_{s'} P \left(s_t = s' | s_0 = s, \pi^* (x)\right) \sum_a \nabla r_{s'a} (x) \pi_{s'a}^* (x) \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_{s', a} P \left(s_t = s', a_t = a | s_0 = s, \pi^* (x)\right) \nabla r_{s'a} (x) \\
&= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t} (x) \; | s_0 = s, \pi^* (x), \mathcal{M}_\tau (x) \right]
\end{aligned}
$$

Absorb P into an expectation

$$
\begin{aligned}
\nabla V_s^* (x) &= \left(I - \nabla_v \varphi \left(x, V^* (x)\right)\right)_s^{-1} \nabla_x \varphi \left(x, V^* (x)\right) \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_{s'} P \left(s_t = s' | s_0 = s, \pi^* (x)\right) \sum_a \nabla r_{s'a} (x) \pi_{s'a}^* (x) \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_{s', a} P \left(s_t = s', a_t = a | s_0 = s, \pi^* (x)\right) \nabla r_{s'a} (x) \\
&= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_x r_{s_t a_t} (x) \; | s_0 = s, \pi^* (x), \mathcal{M}_\tau (x) \right]
\end{aligned}
$$

☺ Estimate via fully first-order information!

Circumvent complicated matrix computations

Hyper-objective

$$\begin{aligned}\phi(x) &= \mathbb{E}_{d_i \sim \rho(d;\pi^*(x))}\left[l\left(d_1, d_2, ..., d_I; x\right)\right] \\ &= \sum_{(d_1,\ldots,d_I)} l\left(d_1, d_2, ..., d_I; x\right) \Pi_{i=1}^{I} P\left(d_i; \pi^*\left(x\right)\right)\end{aligned}$$

Consistency condition

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}: \quad \pi^*_{sa}(x) = \exp\left(\tau^{-1}\left(Q^*_{sa}\left(x\right) - V^*_s\left(x\right)\right)\right)$$

Log probability trick

$$\begin{aligned}&\sum_{(d_1,\ldots,d_I)} l\left(d_1, d_2, ..., d_I; x\right) \nabla \Pi_{i=1}^{I} P\left(d_i; \pi^*\left(x\right)\right) \\ &= \mathbb{E}_{d_i \sim \rho(d;\pi^*(x))}\left[l\left(d_1, d_2, ..., d_I; x\right) \nabla\left(\sum_i \log P\left(d_i; \pi^*\left(x\right)\right)\right)\right] \\ &= \tau^{-1}\mathbb{E}_{d_i \sim \rho(d;\pi^*(x))}\left[l\left(d_1, d_2, ..., d_I; x\right)\left(\sum_i \sum_h \nabla\left(Q^*_{s_h^i a_h^i}\left(x\right) - V^*_{s_h^i a_h^i}\left(x\right)\right)\right)\right]\end{aligned}$$

Second visit to hyper-gradient

Model-free hyper-gradient

$$\nabla\phi(x) = \mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[\nabla l(d_1, d_2, ..., d_I; x)\right]$$
$$+ \tau^{-1}\mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[l(d_1, d_2, ..., d_I; x)\left(\sum_i\sum_h \nabla\left(Q^*_{s_h^i a_h^i}(x) - V^*_{s_h^i}(x)\right)\right)\right]$$

Second visit to hyper-gradient

Model-free hyper-gradient

$$\nabla\phi(x) = \mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[\nabla l\left(d_1, d_2, ..., d_I; x\right)\right]$$
$$+ \tau^{-1}\mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[l\left(d_1, d_2, ..., d_I; x\right)\left(\sum_i\sum_h \nabla\left(Q^*_{s_h^i a_h^i}(x) - V^*_{s_h^i}(x)\right)\right)\right]$$

☺ Estimate $\nabla\phi(x)$ via sampling first-order information!

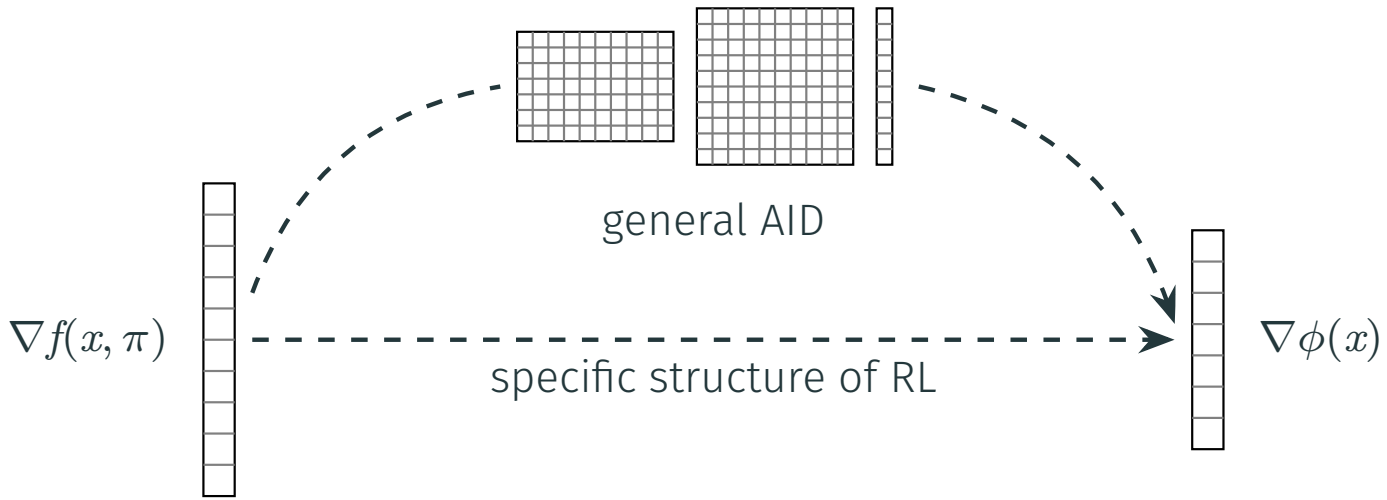
Second visit to hyper-gradient

Model-free hyper-gradient

$$\nabla\phi(x) = \mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[\nabla l(d_1, d_2, ..., d_I; x)\right]$$
$$+ \tau^{-1}\mathbb{E}_{d_i\sim\rho(d;\pi^*(x))}\left[l(d_1, d_2, ..., d_I; x)\left(\sum_i\sum_h \nabla\left(Q^*_{s_h^i a_h^i}(x) - V^*_{s_h^i}(x)\right)\right)\right]$$

☺ Estimate $\nabla\phi(x)$ via sampling first-order information!

Review the hyper-gradient



Construction of estimators

Step 1: evaluating the implicit differentiations

$$\widetilde{\nabla} V_s(x_k, \pi_k) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \;\middle|\; s_0 = s, \pi_k, \mathcal{M}_\tau\,(x_k)\right]$$

$$\widetilde{\nabla} Q_{sa}(x_k, \pi_k) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \nabla r_{s_t, a_t}(x_k) \;\middle|\; s_0 = s, a_0 = a, \pi_k, \mathcal{M}_\tau\,(x_k)\right]$$

Step 2: absorbing them into the sampling process

$$\widetilde{\nabla}\phi(x_k, \pi_k) = \mathbb{E}_{d_i \sim \rho(d;\pi_k)}\left[\nabla l\left(d_1, d_2, ..., d_I; x_k\right)\right]$$
$$+ \tau^{-1}\mathbb{E}_{d_i \sim \rho(d;\pi_k)}\left[l\left(d_1, d_2, ..., d_I; x_k\right)\left(\sum_i \sum_h \widetilde{\nabla}\left(Q_{s_h^i a_h^i} - V_{s_h^i}\right)(x_k, \pi_k)\right)\right]$$

Model-free Soft BiRL algorithm: SoBiRL

Algorithm SoBiRL

Input: iteration number K, step size β, initialization x_1, π_0, accuracy ϵ

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

2: Solve the lower-level problem approximately with initial guess π_{k-1} to get π_k such that $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon$

3: Compute the approximate hyper-gradient $\widetilde{\nabla}\phi(x_k, \pi_k)$

4: Implement an inexact hyper-gradient descent step

$$x_{k+1} = x_k - \beta\widetilde{\nabla}\phi(x_k, \pi_k)$$

5: **end for**

Output: (x_{K+1}, π_{K+1})

Model-free Soft BiRL algorithm: SoBiRL

Algorithm SoBiRL

Input: iteration number K, step size β, initialization x_1, π_0, accuracy ϵ

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

2: Solve the lower-level problem approximately with initial guess π_{k-1} to get π_k such that $\|\pi_k - \pi^*(x_k)\|_2^2 \leq \epsilon$

3: Compute the approximate hyper-gradient $\widetilde{\nabla}\phi(x_k, \pi_k)$

4: Implement an inexact hyper-gradient descent step

$$x_{k+1} = x_k - \beta\widetilde{\nabla}\phi(x_k, \pi_k)$$

5: **end for**

Output: (x_{K+1}, π_{K+1})

☹ Sampling process introduces stochasticity

Main adaptation

- Characterize the bias and variance
- Replace $\widetilde{\nabla}\phi$ with its stochastic counterpart

$$\bar{\nabla}\phi\left(\mathbf{D}_k, \xi_k, \zeta_{\mathbf{k}}; \mathbf{x_k}, \pi_{\mathbf{k}}\right) = \frac{1}{M}\sum_{m=1}^{M} \nabla l\left(\mathbf{d}_k^m; x_k\right) + \frac{1}{\tau M}\sum_{m=1}^{M} l\left(\mathbf{d}_k^m; x_k\right) \sum_i \sum_h \bar{\nabla}\left(Q^{\xi_k}_{s_h^{m,i} a_h^{m,i}} - V^{\zeta_{\mathbf{k}}}_{s_h^{m,i}} \right)(x_k, \pi_k)$$

- Maintain a momentum-instructed h_k for acceleration

$$h_k = \bar{\nabla}\phi_k + (1-\mu_k)\left(h_{k-1} - \bar{\nabla}\phi(\mathbf{D_k}, \xi_{\mathbf{k}}, \zeta_{\mathbf{k}}; \mathbf{x_{k-1}}, \pi_{\mathbf{k-1}})\right)$$

Main adaptation

- Characterize the bias and variance
- Replace $\widetilde{\nabla}\phi$ with its stochastic counterpart

$$\bar{\nabla}\phi\left(\mathbf{D}_k, \xi_k, \zeta_{\mathbf{k}}; \mathbf{x}_{\mathbf{k}}, \pi_{\mathbf{k}}\right) = \frac{1}{M}\sum_{m=1}^{M} \nabla l\left(\mathbf{d}_k^m; x_k\right) + \frac{1}{\tau M}\sum_{m=1}^{M} l\left(\mathbf{d}_k^m; x_k\right) \sum_i \sum_h \bar{\nabla}\left(Q^{\xi_k}_{s_h^{m,i} a_h^{m,i}} - V^{\zeta_{\mathbf{k}}}_{s_h^{m,i}} \right)(x_k, \pi_k)$$

- Maintain a momentum-instructed h_k for acceleration

$$h_k = \bar{\nabla}\phi_k + (1-\mu_k)\left(h_{k-1} - \bar{\nabla}\phi(\mathbf{D_k}, \xi_{\mathbf{k}}, \zeta_{\mathbf{k}}; \mathbf{x_{k-1}}, \pi_{\mathbf{k-1}})\right)$$

☹ Misalignment: distribution of (D_k, ξ_k, ζ_k) relies on π_k

Theoretical Analysis

Assumptions

Upper-level assumptions

- (Model-based) $\nabla f(x, \pi)$ is L_f-Lipschitz, and $\|\nabla_\pi f(x, \pi^*(x)\|_2 \leq C_{f\pi}$
- (Model-free) $l(d_1, d_2, \ldots, d_I; x)$ is L_l-Lipschitz, $\nabla l(d_1, d_2, \ldots, d_I; x)$ is L_{l1}-Lipschitz, and $|l(d_1, d_2, \ldots, d_I; x)| \leq C_l$.

Lower-level assumptions

- For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|r_{sa}(x)| \leq C_r$, $\|\nabla r_{sa}(x)\|_2 \leq C_{rx}$, and $\nabla r_{sa}(x)$ is L_r-Lipschitz

Assumptions

Upper-level assumptions

- (Model-based) $\nabla f(x, \pi)$ is L_f-Lipschitz, and $\|\nabla_\pi f(x, \pi^*(x)\|_2 \leq C_{f\pi}$
- (Model-free) $l(d_1, d_2, \ldots, d_I; x)$ is L_l-Lipschitz, $\nabla l(d_1, d_2, \ldots, d_I; x)$ is L_{l1}-Lipschitz, and $|l(d_1, d_2, \ldots, d_I; x)| \leq C_l$.

Lower-level assumptions

- For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|r_{sa}(x)| \leq C_r$, $\|\nabla r_{sa}(x)\|_2 \leq C_{rx}$, and $\nabla r_{sa}(x)$ is L_r-Lipschitz

☺ Only require first-order regularity!

Convergence analysis (model-based)

Theorem (model-based)

In M-SoBiRL, with constant step sizes β, ξ, and the inner iteration number $N \sim \mathcal{O}(1)$, the iterates $\{x_k\}$ satisfy

$$\frac{1}{K}\sum_{k=1}^{K} \|\nabla \phi(x_k)\|_2^2 = \mathcal{O}\left(\frac{1}{K}\right)$$

Development of hyper-gradient

Proposition (hyper-gradient)

For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $x_1, x_2 \in \mathbb{R}^n$,

$$\begin{aligned}
\|\nabla Q^*_{sa}(x_1) - \nabla Q^*_{sa}(x_2)\|_2 &\leq L_{V1} \|x_1 - x_2\|_2 \\
\|\nabla V^*_s(x_1) - \nabla V^*_s(x_2)\|_2 &\leq L_{V1} \|x_1 - x_2\|_2 \\
\|\nabla \phi(x_1) - \nabla \phi(x_2)\|_2 &\leq L_\phi \|x_1 - x_2\|_2
\end{aligned}$$

with

$$L_{V1} = \frac{(1+\gamma) C_{rx} L_\pi \sqrt{|\mathcal{A}|}}{(1-\gamma)^2} + \frac{L_r}{1-\gamma}$$

$$L_\phi = L_{l1} + L_l H I L_\pi \sqrt{|\mathcal{A}|} + 2\tau^{-1} H I \left(C_l L_{V_1} + \frac{C_{rx}}{1-\gamma} L_l \right) + \frac{2H^2 I^2 C_l C_{rx} L_\pi}{\tau (1-\gamma)} \sqrt{|\mathcal{A}|}$$

Convergence analysis (model-free)

Theorem (model-free)

In SoBiRL, with a constant step size $\beta < \frac{1}{2L_\phi}$, the iterates satisfy

$$\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla\phi\left(x_k\right)\right\|^2 \leq \frac{\phi\left(x_1\right)-\phi^*}{K\left(\frac{\beta}{2}-\beta^2 L_\phi\right)} + \frac{1+2\beta L_\phi}{1-2\beta L_\phi} L_{\widetilde{\phi}}^2\,\epsilon$$

Theorem (stochastic)

In Stoc-SoBiRL, with appropriate sampling configurations, the iterates satisfy

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\phi(x_k)\right\|_2^2\right] = \mathcal{O}\left(\frac{\log K}{K^{2/3}}\right) = \widetilde{\mathcal{O}}\left(K^{-\frac{2}{3}}\right)$$

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(\log\epsilon^{-1}\right)$	1st+2nd
PBRL	Deter.	$\mathcal{O}(\lambda\epsilon^{-1})$	$\mathcal{O}\left(\log\lambda^2\epsilon^{-1}\right)$	1st
HPGD	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}\left(\log\epsilon^{-1}\right)$	1st
M-SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(1\right)$	1st
SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}\left(\log\epsilon^{-1}\right)$	1st
Stoc-SoBiRL	Stoch.	$\widetilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}\left(\log\epsilon^{-1}\right)$	1st

Numerical Experiments

RLHF problem

Goal: learn the intrinsic reward model and the optimal policy
only based on preference labels $y \in \{0, 1\}$

Problem formulation:

$$\begin{aligned} \min_{x} \quad & \mathbb{E}_{y, d_1, d_2 \sim \rho(d; \pi^*(x))} \left[l_h \left(d_1, d_2, y; x \right) \right] \\ \text{s. t.} \quad & \pi^* \left(x \right) = \arg\min_{\pi} - V^{\pi}_{\mathcal{M}_\tau(x)} \left(\rho \right) \end{aligned}$$

Environments: Atari games and Mujoco environments

Default Settings

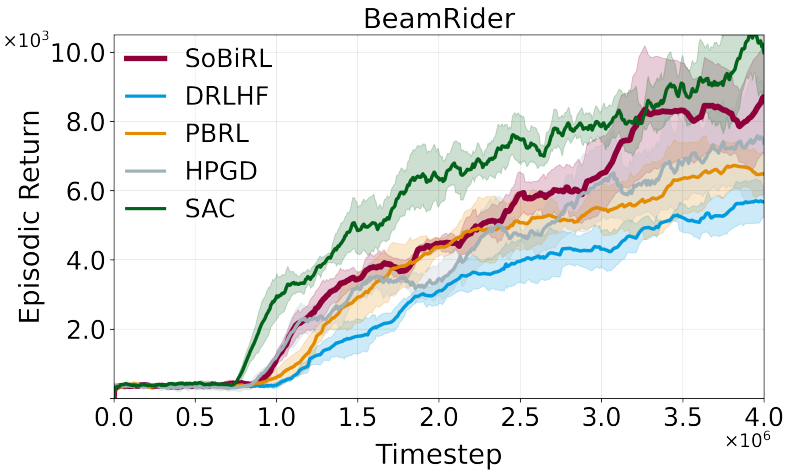
Compared methods

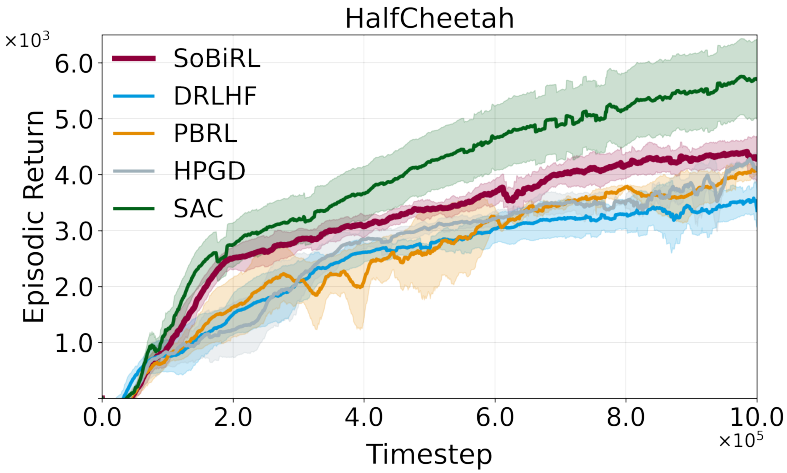
- DRLHF [Christiano et al., 2017]: alternating minimization of $\mathbb{E}[l_h]$ and $-V^{\pi}_{\mathcal{M}_\tau(x)}$
- PBRL [Shen et al., 2024]: penalty-based method
- HPGD [Thoma et al., 2024]: stochastic bilevel RL method
- SAC (baseline) [Haarnoja et al., 2018]: soft actor-critic algorithm given the ground-truth reward

Running platform

- Intel® Xeon® Gold 6330 CPUs & NVIDIA A800 GPU
- Python 3.8.0 + Pytorch 1.13.1 + Stable-Baselines 3

Numerical results





- SoBiRL is comparable to the baseline SAC
- SoBiRL obtains a higher return than other bilevel methods
- The accuracy of preference prediction of DRLHF achieves approximately 85%, while it hovers around 54% for SoBiRL

Third visit to hyper-gradient

Update rule of DRLHF [Christiano et al., 2017] alternates between minimizing $\mathbb{E}[l_h]$ and learning π

$$\mathbf{M}\times: x^+ = x - \beta \mathbb{E}_{d_i \sim \rho(d;\pi^*(x))} \left[\nabla l(d_1, d_2, ..., d_I; x)\right]$$
$$\mathbf{N}\times: \pi^+ = \pi + \nabla_\pi V^\pi_{\mathcal{M}_\tau(x)}(\rho)$$

Update rule of SoBiRL (ours)

$$\mathbf{1}\times: x^+ = x - \beta \nabla \phi(x)$$
$$\mathbf{N}\times: \pi^+ = \pi + \nabla_\pi V^\pi_{\mathcal{M}_\tau(x)}(\rho)$$

$$\nabla\phi(x) = \mathbb{E}_{d_i \sim \rho(d;\pi^*(x))} \left[\nabla l(d_1, d_2, ..., d_I; x)\right]$$
$$+ \tau^{-1} \mathbb{E}_{d_i \sim \rho(d;\pi^*(x))} \left[l(d_1, d_2, ..., d_I; x) \left(\sum_i \sum_h \nabla \left(Q^*_{s_h^i a_h^i}(x) - V^*_{s_h^i}(x) \right) \right) \right]$$

☺ $\nabla\phi(x)$ integrates exploitation and exploration!

Conclusions and perspectives

Contributions

- Fully first-order hyper-gradient
- AID-based RL algorithm without lower-level convexity
- Stochastic extension
- Enhanced convergence results

Future works

- Extension to general strongly-convex regularization
- General settings without regularization

References

- **Yan Yang**, Bin Gao, Ya-xiang Yuan. *LancBiO: dynamic Lanczos-aided bilevel optimization via Krylov subspace.*
  ICLR (2025)
- **Yan Yang**, Bin Gao, Ya-xiang Yuan. *Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity.*
  AISTATS (2025)

- Code is publicly available from https://github.com/UCAS-YanYang

Thanks for your attention!

yangyan@amss.ac.cn